



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

AUTOMATICKÉ POLOHOVÁNÍ ZPĚTNÉHO ZRCÁTKA

AUTOMATIC POSITIONING OF THE REAR VIEW MIRROR

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Marek Návara

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Peter Honec, Ph.D.

BRNO 2016

Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Marek Návara

ID: 146913

Ročník: 2

Akademický rok: 2015/16

NÁZEV TÉMATU:

Automatické polohování zpětného zrcátka

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce bude navrhnout a implementovat systém pro automatické polohování zpětného zrcátka automobilu v závislosti na poloze očí řidiče.

1. Proveďte rešerši metod pro 3D rekonstrukci scény.
2. Navrhněte vhodné HW vybavení (krokové motory, kamera, výpočetní jednotka).
3. Implementujte algoritmy detekce obličeje a pozice očí řidiče.
4. Navrhněte a implementujte řízení polohy zrcátek.

Ověřte a zhodnoťte.

DOPORUČENÁ LITERATURA:

Hlaváč, Šonka, Počítačové vidění.

Šonka, Hlaváč, Boyle - IMAGE PROCESSING, ANALYSIS, AND MACHINE VISION,

Termín zadání: 8.2.2016

Termín odevzdání: 16.5.2016

Vedoucí práce: Ing. Peter Honec, Ph.D.

Konzultant diplomové práce:

doc. Ing. Václav Jirsík, CSc., předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato práce řeší návrh funkčního zařízení, které bude schopno automaticky polohovat zpětná zrcátka dle polohy obličeje řidiče. Měření pozice obličeje vůči zrcátku zajišťuje stereovidění dvou webkamer. Zařízení je postaveno na počítači Raspberry Pi 2, ke kterému byla navržena rozšiřující deska.

Vytvořený prototyp dokáže sledovat nastavený výhled v zrcátku s přesností do 7 cm (v horizontálním směru do 5cm) v úrovni zadního rohu automobilu. Výsledky této práce ověřují návrh automatického polohování zpětného zrcátka a lze na nich založit konkrétní implementaci zařízení do automobilu.

Klíčová slova

zpětné zrcátko, automatické polohování zrcátka, zpracování obrazu, webkamera, stereovize, Raspberry Pi 2, OpenCV

Abstract

This thesis solves design of functional device that will be able to automatically positioning back view mirrors according to the positio of driver face. Measuring position of the face provides stereovision of two webacams. The device is based on a computer Raspberry Pi 2 with designed expansion board.

The created prototype can follow set view int the mirror with accuracy up to 7 cm (horizontally up to 5cm) in level of rear corner of a car. The results of this project validate design of automatic positioning mirror and it can be basis for specific implementations of the device in car.

Keywords

rear view mirror, automatic positioning of mirror, image processing, webcam, stereovision, Raspberry Pi 2, OpenCV

Bibliografická citace:

NÁVARA, M. *Automatické polohování zpětného zrcátka*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 73s. Vedoucí diplomové práce byl Ing. Peter Honec, Ph.D.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Automatické polohování zpětného zrcátka jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **16. května 2016**

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Peteru Honcovi, Ph.D. za účinné odborné vedení, ochotu a cenné rady při zpracování mé diplomové práce.

V Brně dne: **16. května 2016**

.....
podpis autora

Obsah

1	Úvod	11
2	Zpracování obrazu	12
2.1	Digitální interpretace obrazu	12
2.2	Osvětlení a snímání scény	13
2.3	Předzpracování obrazu	14
2.4	Segmentace	16
2.5	Klasifikace	16
3	Detekce obličeje	18
3.1	Znalostní metody	18
3.2	Metody neměnných prvků	19
3.3	Porovnání se šablonou	19
3.4	Metody založené na vzhledu	19
3.5	Detekce dle barvy kůže	19
3.6	Detektor Viola-Jones	20
4	Optické měření pozice ve 3D	24
4.1	Přehled metod pro optické měření pozice ve 3D	24
4.2	Stereovize	27
5	Rozbor úlohy	29
5.1	Popis scény	29
5.2	Funkce zařízení	30
5.3	Měření polohy obličeje	30
6	Návrh hardware	36
6.1	Kamera	36
6.2	Zpětné zrcátko	38
6.3	Výpočetní jednotka	40
6.4	Rozšiřující deska	42
6.5	Návrh stojanu	46
7	Softwarová implementace	47
7.1	ATMega8	47
7.2	Instalace Raspbian	49
7.3	Použité knihovny	49
7.4	Spuštění programu	50
7.5	Architektura aplikace	51

7.6	Nastavení kamer.....	52
7.7	Předzpracování obrazu	53
7.8	Detektor obličeje	54
7.9	Výpočet natočení zrcátka	56
8	Vyhodnocení chování zařízení	60
8.1	Úspěšnost detekce obličeje	60
8.2	Přesnost měření pozice pomocí stereovize.....	62
8.3	Přesnost automatického nastavení výhledu v zrcátku	63
9	Závěr.....	65

Seznam obrázků

OBRÁZEK 1: ŘETĚZEC ZPRACOVÁNÍ OBRAZU. PŘEVZATO Z [3.]	12
OBRÁZEK 2: DIGITÁLNÍ REPREZENTACE BAREVNÉHO OBRÁZKU (RGB)	12
OBRÁZEK 3: PŘÍKLAD RŮZNÝCH OSVĚTLENÍ. A) PŘÍMÉ OSVĚTLENÍ, B) ZADNÍ OSVĚTLENÍ. PŘEVZATO Z [8.]	13
OBRÁZEK 4: PŘÍKLAD FILTRACE POUŽITÍM GAUSSOVA FILTRU (VELIKOST 7×7 , $\sigma=2$)	14
OBRÁZEK 5: PŘÍKLAD DETEKCE HRAN (SOBELŮV OPERÁTOR)	14
OBRÁZEK 6: GEOMETRICKÉ ZKRESLENÍ. A) RADIÁLNÍ, B) TANGENCIÁLNÍ, [3.]	15
OBRÁZEK 7: PŘÍKLAD JASOVÉ TRANSFORMACE. A) ORIGINÁLNÍ SNÍMEK, B) ZMĚNA JASU, C) ZMĚNA KONTRASTU	15
OBRÁZEK 8: KLASIFIKÁTOR	17
OBRÁZEK 9: SCHÉMA PERCEPTRONU, [1.]	17
OBRÁZEK 10: ROZLOŽENÍ BAREV KŮŽE V MODELECH RGB A $YCbCr$. PŘEVZATO Z [5.]	20
OBRÁZEK 11: HAAROVY PŘÍZNAKY	21
OBRÁZEK 12: INTEGRÁLNÍ OBRAZ. HODNOTA BODU $I_i(x,y)$ SE ROVNÁ SUMĚ PIXELŮ PŮVODNÍHO OBRAZU V ŠEDÉM OBDÉLNÍKU.	22
OBRÁZEK 13: PŘÍKLAD VÝPOČTU SUMY PIXELŮ Z INTEGRÁLNÍHO OBRAZU.	22
OBRÁZEK 14: PŘÍKLAD 2-D AKTIVNÍ TRIANGULACE, [8.]	24
OBRÁZEK 15: PŘÍKLAD SAMOKALIBRUJÍCÍ PASIVNÍ TRIANGULACE. PŘEVZATO Z [8.]	25
OBRÁZEK 16: VYUŽITÍ INTERFEROMETRU PRO MĚŘENÍ POVRCHU OBJEKTU. A) PRINCIPÁLNÍ SCHÉMA SNÍMAČE, B) SKVRNOVÝ INTERFEROGRAM, C) VÝSLEDNÝ MODEL ZMĚŘENÉHO POVRCHU (VYŽADUJE ALESPŮŇ 3 MĚŘENÍ S ROZDÍLNOU VZDÁLENOSTÍ SENZORU OD OBJEKTU). [8.]	26
OBRÁZEK 17: STEREOVIZE – DVĚ KAMERY S PARALELNÍMI OPTICKÝMI OSAMI.[9.]	27
OBRÁZEK 18: POUŽITÍ IR PŘÍSVITU VE TMĚ V INTERIÉRU AUTOMOBILU.	29
OBRÁZEK 19: DIAGRAM ZNÁZORŇUJÍCÍ ZÁKLADNÍ PRINCIP ZAŘÍZENÍ	30
OBRÁZEK 20: POZICE HLAVY ŘIDIČE V AUTOMOBILU	31
OBRÁZEK 21: ZNÁZORNĚNÍ PŘEPOČÍTANÉ CHYBY E POLOHY ŘIDIČE ZE ZNÁMÝCH VZDÁLENOSTÍ L_0 A I . ČERVENĚ JE VYZNAČEN PŘEPOČET CHYBY DO SOUŘADNIC X A Z	32
OBRÁZEK 22: CITLIVOST STEREOVIZE	33
OBRÁZEK 23: GEOMETRIE VÝPOČTU VELIKOSTI BÁZE STEREOVIZE	34
OBRÁZEK 24: VLIV VELIKOSTI BÁZE STEREOVIZE NA VÝSLEDNOU VÝSEČ PRO VYBRANÉ KAMERY	35
OBRÁZEK 25: SROVNÁNÍ SNÍMKŮ POŘÍZENÝCH S FILTREM A BEZ FILTRU. HORNÍ ŘADA ODPOVÍDÁ SNÍMKŮM BEZ FILTRU, DOLNÍ SNÍMKŮM S FILTREM. V LEVÉM SLOUPCI JSOU PAK SNÍMKY POŘÍZENÉ V NOCI A V PRAVÉM PŘI DENNÍM SVĚTLE.	37
OBRÁZEK 26: GRAF ZÁVISLOSTI RELATIVNÍ PROPUSTNOSTI NA VLNOVÉ DÉLCE SVĚTLA	37
OBRÁZEK 27: FOTKA POŘÍZENÉHO ELEKTRICKÉHO ZRCÁTKA S PAMĚTÍ PRO VŮZ ŠKODA OCTAVIA	38
OBRÁZEK 28: SCHÉMA ZAPOJENÍ POŘÍZENÉHO ZRCÁTKA	38
OBRÁZEK 29: POUŽITÉ OVLÁDÁNÍ ZRCÁTEK URČENÉ PRO AUTOMOBIL ŠKODA OCTAVIA	39
OBRÁZEK 30: SCHÉMA VNITŘNÍHO ZAPOJENÍ OVLADAČE ZRCÁTEK A PŘIPOJENÍ K VYHODNOCOVAČÍMU OBVODU	39

OBRÁZEK 31: VHODNÉ EMBEDDED MODULY. A) RASPBERRY PI 2, B) BEAGLEBONE BLACK, C) ODROID-C1+	40
OBRÁZEK 32: POPIS PINŮ A JEJICH FUNKCÍ GPIO KONEKTORU RASPBERRY PI 2.....	41
OBRÁZEK 33: ZNÁZORNĚNÍ POUŽITÝCH PINŮ KONEKTORU GPIO A JEJICH ZAPOJENÍ	42
OBRÁZEK 34: VYROBENÁ ROZŠÍŘUJÍCÍ DESKA.....	42
OBRÁZEK 35: SCHÉMA NAPÁJECÍ ČÁSTI	43
OBRÁZEK 36: SCHÉMA ZAPOJENÍ ATMEGA8 NA ROZŠÍŘUJÍCÍ DESCE.....	44
OBRÁZEK 37: ZAPOJENÍ DVOJITÉHO H-MŮSTKU PRO ŘÍZENÍ NATOČENÍ ZRCÁTKA.....	45
OBRÁZEK 38: KONSTRUKČNÍ 3D NÁVRH STOJANU PRO TESTOVÁNÍ ZAŘÍZENÍ.....	46
OBRÁZEK 39: ALGORITMUS AD PŘEVODU A ODESÍLÁNÍ DAT PŘES I2C.....	47
OBRÁZEK 40: FUNKCE JEDNOTLIVÝCH BITŮ V PROMĚNNÉ JOYSTICK.....	48
OBRÁZEK 41: POSTUP AUTOMATICKÉHO SPUŠTĚNÍ PŘI ZAPNUTÍ RASPBERRY	50
OBRÁZEK 42: VÝVOJOVÝ DIAGRAM ZNÁZORŇUJÍCÍ ARCHITEKTURU VYTVOŘENÉ APLIKACE	51
OBRÁZEK 43: POSTUP I2C KOMUNIKACE.....	52
OBRÁZEK 44: POUŽITÝ POSTUP PŘEDZPRACOVÁNÍ OBRAZU	53
OBRÁZEK 45: ALGORITMUS IMPLEMENTOVANÉ DETEKCE OBLIČEJE	55
OBRÁZEK 46: POSTUP VÝPOČTU AUTOMATICKÉHO NATOČENÍ ZRCÁTKA	56
OBRÁZEK 47: VYZNAČENÍ ZVOLENÝCH SOUŘADNICOVÝCH SYSTÉMŮ POUŽITÝCH PRO VÝPOČTY.....	56
OBRÁZEK 48: VLIV FILTRACE NA VÝSLEDNOU POZICI OBLIČEJE	57
OBRÁZEK 49: ZMĚNA POOTOČENÍ ZRCÁTKA PŘI ZMĚNĚ POLOHY OBLIČEJE	58
OBRÁZEK 50: UKÁZKA SNÍMKŮ Z VYTVOŘENÉ TESTOVACÍ SADY. SNÍMKY JSOU PO PRŮCHODU ALGORITMEM S VYZNAČENÝM DETEKOVANÝM OBLIČEJEM. SNÍMEK VLEVO JE POŘÍZEN ZA DENNÍHO SVĚTLA, SNÍMEK VPRAVO V NOCI	60
OBRÁZEK 51: VÝPOČTENÉ ABSOLUTNÍ CHYBY MĚŘENÍ POLOHY V ZÁVISLOSTI NA MĚŘENÝCH SOUŘADNICÍCH. V LEVÉ ČÁSTI JE ZOBRAZENA V MĚŘENÍ X-OVÉ SOUŘADNICE. VPRAVO JE VYKRESLENA ZÁVISLOST CHYBY MĚŘENÍ SOUŘADNICE Y.	62
OBRÁZEK 52: CHYBA MĚŘENÍ VZDÁLENOSTI OBLIČEJE OD KAMERY POMOCÍ STEREOVIZE V ZÁVISLOSTI NA NASTAVENÝCH SOUŘADNICÍCH X A Y	63
OBRÁZEK 53: ROZLOŽENÍ CHYB VÝHLEDU VE VZDÁLENOSTI 2 M PŘI AUTOMATICKÉM NATÁČENÍ ZRCÁTKA. PROVEDENO 30 MĚŘENÍ PRO RŮZNÉ POLOHY OBLIČEJE V INTERVALECH UVAŽOVANÝCH V AUTOMOBILU. ČERVENĚ JSOU VYZNAČENY BODY, KTERÉ NEVYHOVUJÍ STANOVENÉ CHYBĚ 5 CM VE VZDÁLENOSTI 4 METRŮ.....	64

1 ÚVOD

Aktuálním trendem v automobilovém průmyslu je implementace podpůrných systémů pro řidiče. Prvním typem jsou systémy zvyšující bezpečnost jízdy. Především již standardní systémy ABS, ASR, či ESP, ale také nově implementované systémy hlídání mrtvého úhlu, automatická detekce značek apod. Druhá kategorie podpůrných systémů pro řidiče zvyšuje komfort. Může se jednat o v dnešní době typické elektrické ovládání oken, klimatizaci, či zatím méně časté automatické rozsvěcení dálkových světel, nebo multimediální palubní počítač umožňující nejen zobrazení stavu vozidla, vestavěnou navigaci, ale i multimediální přehrávač, nebo připojení na internet.

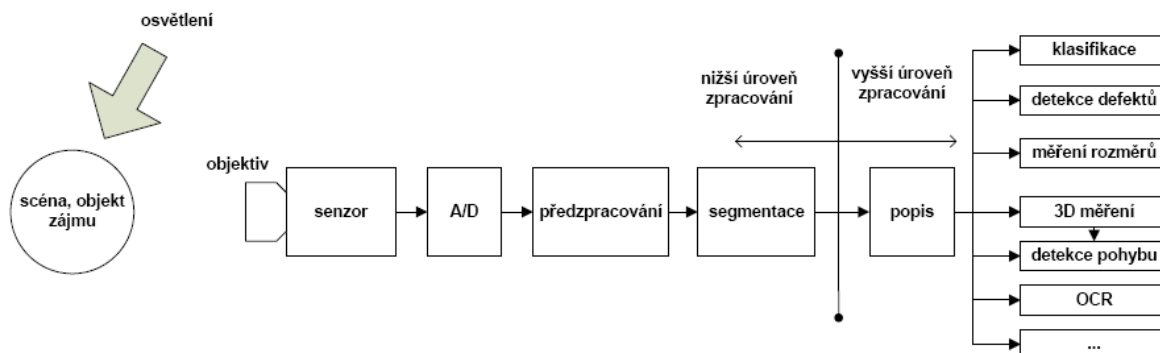
Tento projekt se zabývá jedním ze systémů zvyšujících komfort, který není běžně používán. Jedná se o automatické natáčení zpětných zrcátek dle polohy řidiče. Zrcátka se automaticky natáčí tak, aby řidič v zrcátku viděl stále stejnou scénu. Tento systém by měl být schopný reagovat na krátkodobé posunutí hlavy řidiče (například pokud se nahne dopředu), ale také nahradit manuální nastavení zrcátek v případě, kdy do auta usedne jiný řidič.

Navrhovaný systém bude využívat zpracování obrazu, které má veliký potenciál využití v automobilovém průmyslu. Již nyní lze narazit na systémy detekce únavy řidiče, které jsou založeny právě na analýze obrazu řidiče. V době rozšiřujících se možností výpočetních jednotek se stává zpracování obrazu jedním z důležitých přístupů ke snímání.

Pozice obličeje bude muset být měřena v trojrozměrném prostoru (3D), aby bylo možné detekovat změnu pozice obličeje vůči bočním zrcátkům. V práci bude provedena analýza snímané scény a následně bude proveden výběr odpovídajícího hardwaru. V další části budou navrženy algoritmy detekce obličeje a následného automatického natáčení zrcátka. Algoritmy budou implementovány do vhodné výpočetní jednotky. Po vytvoření funkčního zařízení bude otestována jeho funkčnost na vzorových situacích a zhodnocena kvalita sledování nadefinovaného výhledu v zrcátku.

2 ZPRACOVÁNÍ OBRAZU

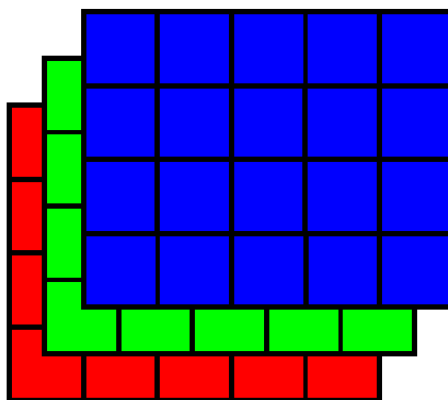
V následující kapitole jsou stručně popsány jednotlivé části zpracování obrazu. Zpracováním obrazu nazýváme celý řetězec kroků od nasvícení objektu, přes snímání, předzpracování až po segmentaci a popisu objektů.



Obrázek 1: Řetězec zpracování obrazu. Převzato z [3.].

2.1 Digitální interpretace obrazu

Šedotónový digitalizovaný obraz po provedení vzorkování a kvantizace můžeme ve výpočetní technice uvažovat jako dvourozměrnou matici. Hodnoty prvků matice, vyjadřující jednotlivé pixely, se pohybují nejčastěji v rozsahu celočíselných hodnot 0-255 (8bitová hloubka). U barevného obrazu máme třírozměrnou matici, která obsahuje 3 matice, kde každá matice náleží jedné barevné složce. Nejznámější model RGB, znázorněn na obrázku 2, se skládá z červené (red), zelené (green) a modré (blue) složky. Barva však může být interpretována i jinými barevnými modely. Model CMY se skládá z azurové (Cyan), purpurové (Magenta) a žluté (Yellow). Odlišnou variantou je model YC_bC_r . Tento model obsahuje složky: Y – jas, C_b – modrá chrominanční komponenta a C_r – červená chrominanční komponenta

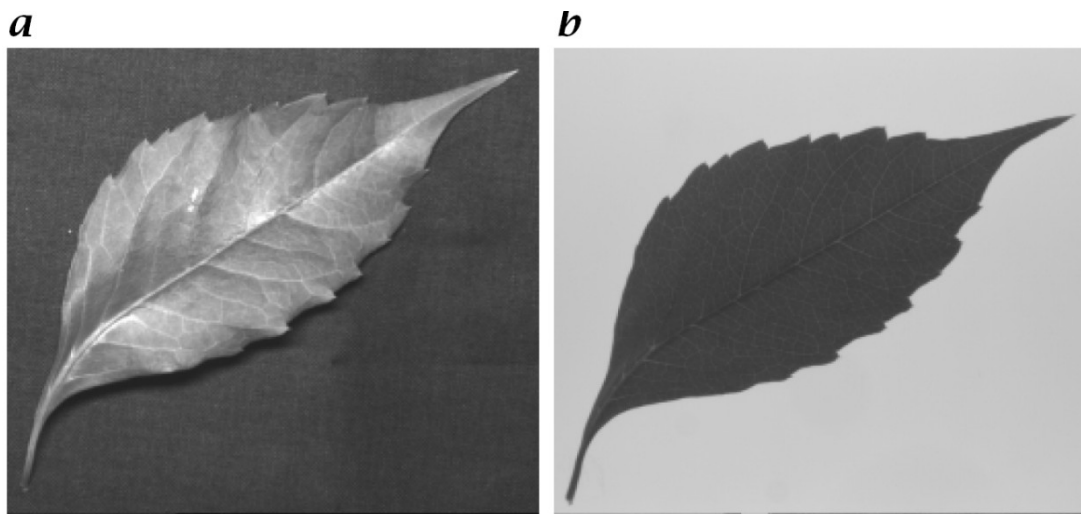


Obrázek 2: Digitální reprezentace barevného obrázku (RGB).

2.2 Osvětlení a snímání scény

První a důležitá část řetězce je osvětlení scény a její následné snímání. V tomto kroku můžeme největší měrou ovlivnit, kolik adekvátních informací získáme pořízením obrazu. Ačkoliv lze v mnoha aplikacích počítačového vidění vynechat přísvit a pracovat pouze s okolními světelnými podmínkami, může být v jiných aplikacích přísvit nutný. A nebo alespoň může znatelně ulehčit následné zpracování obrazu.

Lze použít přísvit různé vlnové délky (IR, viditelné) v závislosti na požadavcích aplikace. Dalším krokem je volba provedení a umístění světelného zdroje. Využívají se bodové zdroje, plošné zdroje, světelné pruhy a mřížky. Záleží také na umístění světelného zdroje. Na některé aplikace se hodí přímé osvětlení, na jiné difuzní, boční či zadní osvětlení.[2.] Přímé osvětlení se používá k zvýraznění textury objektu (Obrázek 3.a) a nebo tam, kde nejsou speciální požadavky na osvětlení, ale přísvit je vyžadován. Zadní přísvit se používá především k zvýraznění tvaru neprůsvitných objektů, k detekci přítomnosti malých objektů a nebo k diagnostice poruch poloprůsvitných objektů.



Obrázek 3: Příklad různých osvětlení. a) přímé osvětlení, b) zadní osvětlení. Převzato z [8.]

Vlastnosti nasnímaného obrazu závisí také na použité optice objektivu a dalších přidavných doplňcích. Objektivy ovlivňují zejména zorné pole, zvětšení, rozsah ostření a hloubku ostrosti. Doplňky jsou zejména přidavné filtry a to jak polarizační, tak i filtry horní či dolní propusti. Používán je zejména filtr na odfiltrování infračerveného spektra, který obsahuje každá běžná kamera, či fotoaparát. Druhé používané jsou naopak filtry, které propouští pouze úzké pásmo vlnových délek. Například filtr s pásmovou propustí pro IR světlo vlnové délky kolem 900nm (v závislosti na zdroji světla) se používá pro noční přísvit.

2.3 Předzpracování obrazu

Předzpracování obrazu se používá v důsledku nedokonalého nasnímání vstupního obrazu. Pořízený obraz má nedokonalosti, jako je šum, zkreslení optikou, rozostření, či nevhodné rozložení jasu.

Předzpracování obrazu má především následující úkoly [1.]:

- Odstranění šumu – Filtrace

Šum se vyskytuje v každém reálném snímku vlivem nedokonalosti snímání. Při filtrování se používá nadbytečnosti dat. Filtrování obrazu může být řešeno buď použitím konvoluční masky filtru a nebo filtrováním ve frekvenční oblasti. Nejznámějším filtrem je Gaussův filtr. Ten má normální rozložení masky. Další používané metody jsou průměrování a medián.



Obrázek 4: Příklad filtrace použitím Gaussova filtru (velikost 7x7, $\sigma=2$).

- Detekce hran

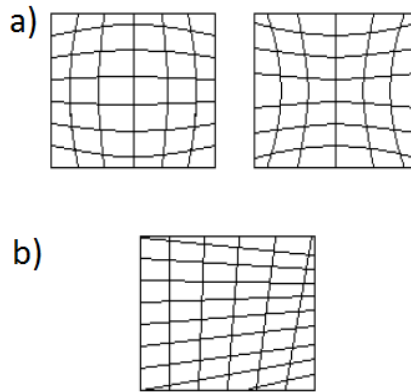
Zdůrazňuje části obrazu, kde se jasová funkce strmě mění. Pro detekci hran se využívají operátory aproximující první derivaci; Robertsův, Sobelův, Prewittův. Tyto operátory nejsou invariantní vůči rotaci – musejí se použít rotované masky pro více směrů [3.]. Invariantní vůči rotaci je například Laplaceův operátor. Dalšími používanými detektory jsou: Cannyho hranový detektor a operátor LoG.



Obrázek 5: Příklad detekce hran (Sobelův operátor).

- Geometrická transformace

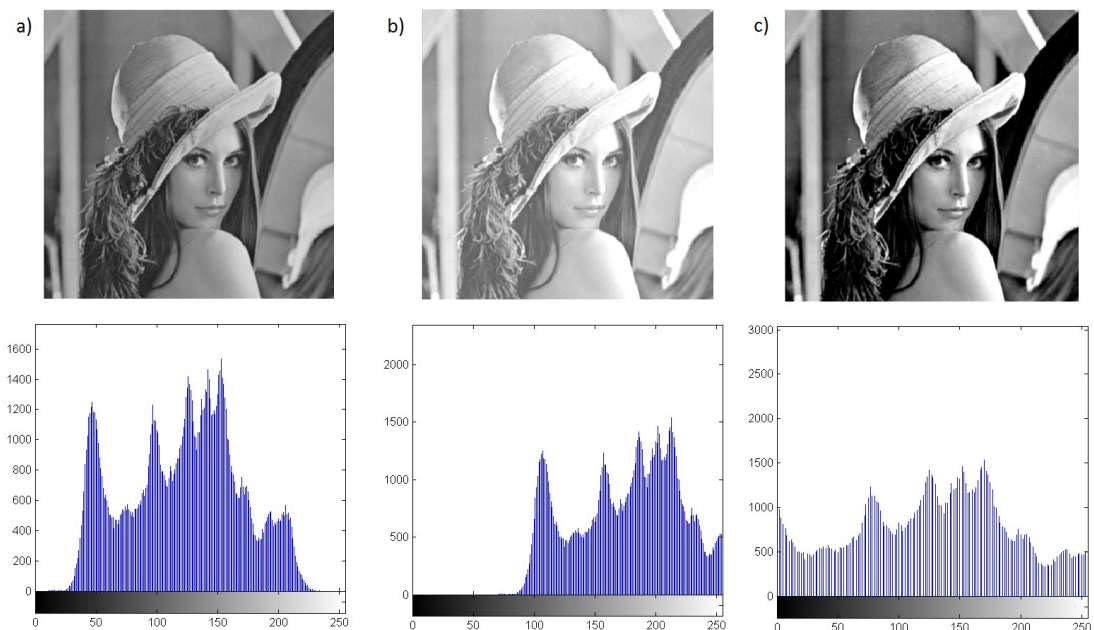
Má za úkol odstranit geometrické zkreslení. To může vznikat vlivem vady optické soustavy (radiální zkreslení), nedokonalostí umístění senzoru (tangenciální zkreslení), vzájemnou polohou senzoru a objektu (zkosení, otočení, změna měřítka).



Obrázek 6: Geometrické zkreslení. a) radiální, b) tangenciální, [3.]

- Jasová korekce

Jedním z míst, kde se jasová korekce používá je odstranění nerovnoměrné citlivosti snímáče či pro nerovnoměrné osvětlení. Další použití jasové korekce je pro úpravy obrazu k dalšímu zpracování. Ty používají k jasové transformaci převodní charakteristiku. Mezi takové úpravy patří především: úprava jasu, úprava kontrastu, redukce barev, gama korekce nebo ekvalizace histogramu.



Obrázek 7: Příklad jasové transformace. a) originální snímek, b) změna jasu, c) změna kontrastu

2.4 Segmentace

Segmentace obrazu má za úkol oddělit jednotlivé objekty od sebe a od pozadí a to především z důvodu redukce dat a k přípravě pro popis objektů. Metody používané pro segmentaci dělíme na[1.]:

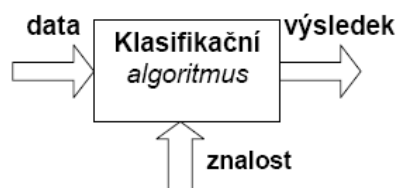
- Segmentace prahováním
Lze použít tam kde objekt a pozadí mají oddělitelné jasové úrovně. Nejjednodušší je prosté prahování, které použije jeden práh na celý obraz. Složitější metody používají vícenásobné prahování nebo tzv. adaptivní prahování, kdy se práh určuje z lokálního okolí. Určení prahu se provádí ze znalosti obrazu, z histogramu, experimentálně nebo procentně.
- Segmentace využívající detekci hran
Metoda vychází z předpokladu, že objekty jsou odděleny hranou. Objekty můžeme oddělit prahováním obrazu hran, sledováním hranice objektu, Houghovou transformací, nebo aktivními konturami.
- Metody orientované na regiony
Mezi metody orientované na regiony patří metoda spojování oblastí, metoda štěpení oblastí či shluková analýza. Nejpoužívanější algoritmy shlukové analýzy jsou Mean-shift a K-means.
- Znalostní metody – srovnání se vzorem
- Hybridní metody – neuronové sítě, morfologické operace

2.5 Klasifikace

Po úspěšné segmentaci jsou již objekty odděleny od sebe, či od pozadí a klasifikace má určit o jaké objekty se jedná. Nejčastěji se používá strojová klasifikace (angl. Patternrecognition), kdy je klasifikátor založen na strojovém učení. Postup této metody má 3 části:

1. Učení – generování modelu pomocí trénovací množiny
2. Verifikace – ověření modelu na datech odlišných od trénovacích
3. Klasifikace – využití naučeného klasifikátoru pro klasifikaci neznámých dat

Klasifikátor je algoritmus, který má na vstupu příznaky a na základě vhodných znalostí je schopen dle těchto příznaků data klasifikovat do výstupních tříd.

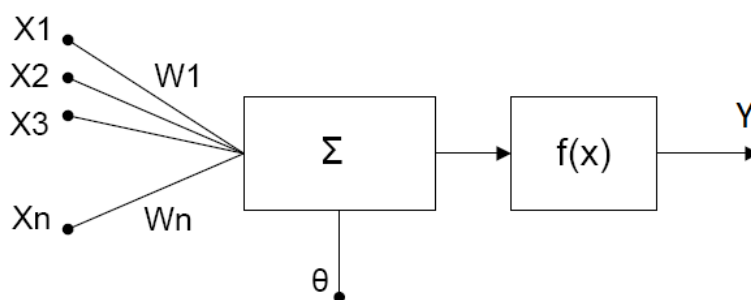


Obrázek 8: Klasifikátor

Důležitá pro správnou klasifikaci je nejen volba klasifikátoru, ale především výběr vhodných příznaků popisujících objekt.

2.5.1 Perceptron

Perceptron je klasifikátor (model), který je inspirován biologickým neuronem v mozku. Jedná se o velmi elementární klasifikátor, u kterého je každý ze vstupů x_i vážený váhou w_i . Parametr θ se nazývá práh neuronu. Funkce $f(x)$ je přenosová funkce neuronu. Na výběr je mnoho typů funkcí a mezi základní patří skoková, sigmoidální a hyperbolický tangens.[1.]



Obrázek 9: Schéma perceptronu, [1.]

$$Y = f\left(\sum_{i=1}^N (w_i \cdot x_i) + \theta\right) \quad (2-1)$$

3 DETEKCE OBLIČEJE

Detekce obličejů má za úkol určit zda se v obraze nachází či nenachází obličej a případně určit polohu a oblast obličejů. Klasifikace, zda v dané oblasti obličej je, nebo není, je náročná vzhledem k velké rozmanitosti lidských obličejů. Proto je důležité zvolit nejen vhodný klasifikátor vzhledem k následnému použití, ale také vhodnou trénovací sadu dat.

Vlivy na detekci obličejů:

- Vzájemná poloha obličejů a kamery, což určuje natočení hlavy: přední pohled, natočení o určitý úhel, nebo z profilu. Vzdálenost mezi obličejem a kamerou ovlivňuje výslednou velikost obličejů v obraze. Detekce musí zvládat různě velké obrazy obličejů.
- Přítomnost méně častých komponent obličejů. Těmi jsou především brýle a vousy, které mohou mít různou velikost, barvu a tvar. Při návrhu detektoru je třeba s nimi počítat.
- Výraz v obličejích. Rozmanitost výrazů v obličejích klade nároky na robustnost klasifikátoru.
- Jasové podmínky při pořizování obrazu významně ovlivňují výsledný obraz obličejů. Tím je myšlena nejen intenzita jasu, ale také rozložení světla, což může způsobovat nerovnoměrné osvětlení.
- Neúplný, či z části zakrytý obličej způsobuje ztrátu části informace.

V současné době existuje mnoho přístupů k detekci obličejů v obraze. Tyto metody pro detekci obličejů lze rozdělit do čtyř hlavních kategorií[6.]:

- Znalostní metody,
- příznakové algoritmy,
- algoritmy založené na porovnávání se šablonou
- a metody založené na vzhledu.

3.1 Znalostní metody

Znalostní metody se zakládají na pravidlech popisujících základní rysy tváře a jejich vzájemné vztahy. Obvykle popisují vzájemnou polohu obličejových rysů, jako například pozici očí vůči sobě a vůči dalším hlavním rysům (nos, pusa). Nejprve probíhá extrakce jednotlivých rysů z obrazu a poté následuje vlastní detekce na základě znalosti základních pravidel. Úskalím této metody je přesný popis pravidel, který by však zahrnoval rozmanitost lidských obličejů.

3.2 Metody neměnných prvků

Nebo též metody invariantních rysů vyhledávají základní prvky obličeje, které lze nalézt při různém natočení, světelných podmínkách, či pozici obličeje. Těmito rysy mohou být obličejové příznaky (oči, ústa, nos apod.) obličejová textura a nebo barva kůže.

3.3 Porovnání se šablonou

Tyto metody jsou založeny na korelaci mezi předdefinovanými vzory a vstupním obrazem. Buďto je vzorem celý obraz a nebo sada vzorů, obsahující vzory pro jednotlivé části a kontury obličeje. Detekce tváře je poté založena na hodnotách korelační funkce, neboli do jaké míry odpovídá vstupní obraz vzorové tváři. Výhodou metody je snadná implementace založena pouze na korelaci. V praxi však není tento přístup dostačující z důvodů různých rozměrů orientace či tvaru částí obličeje. Také při změně rozlišení snímků, velikosti či natočení obličeje nelze použít jednu předdefinovanou sadu.

3.4 Metody založené na vzhledu

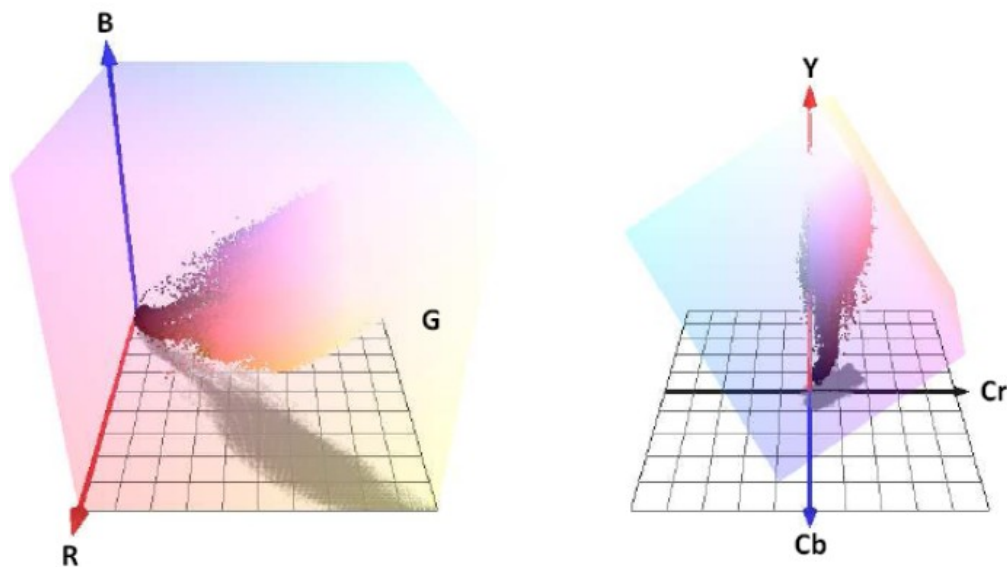
Na rozdíl od porovnání se šablonou, v tomto přístupu nejsou vzory definovány přímo, ale jsou počítány z velkého množství trénovacích dat. Metody jsou založeny na strojovém učení a statistické analýze. Detektory založené na učení lépe klasifikují obličeje v různých podmínkách.

Většinu z v současné době používaných algoritmů nelze zahrnout pouze do jedné z uvedených skupin, ale využívají přednosti z více skupin, aby dosáhli lepších výsledků.

3.5 Detekce dle barvy kůže

Ač všechny výše popsané metody používají monochromatický obraz, je segmentace dle barvy kůže často používanou součástí systémů pro detekci obličeje. Její předností je invariantnost vůči natočení nebo poloze obličeje ve scéně, ale také rychlost zpracování. Používá se pro předzpracování, kdy zužuje oblast pro další vyhodnocení.

Základním principem této metody je fakt, že barva lidské kůže se nachází v určitém podprostoru barevného modelu. Model RGB není ke klasifikaci příliš vhodný, protože barva kůže různých lidí se liší více v jasové složce, než v barevném odstínu. Daleko vhodnějším modelem je $YCbCr$, u kterého je jas definován pouze jednou složkou a segmentovat tedy lze pomocí zbylých dvou složek udávajících chromatičnost. Srovnání rozložení barev kůže v modelech RGB a $YCbCr$ je na Obrázek 10.



Obrázek 10: Rozložení barev kůže v modelech RGB a $YCbCr$. Převzato z[5.]

3.6 Detektor Viola-Jones

Detektor obličejů představený v roce 2001[7.] pracuje se šedotónovým obrazem a je jedním z nejpoužívanějších algoritmů pro detekci nejen obličejů, ale souhrnně objektů. Algoritmus má přednosti v rychlosti a robustnosti ke změně světelných podmínek a velikosti se zachováním dostatečné spolehlivosti. Algoritmus patří mezi metody založené na vzhledu.

Detektor je založen na strojovém učení, kdy vybíráme příznaky popisující objekt pomocí trénovací sady obrazů, která musí obsahovat pozitivní i negativní vzory. Metoda je založena na kombinaci tří základních částí:

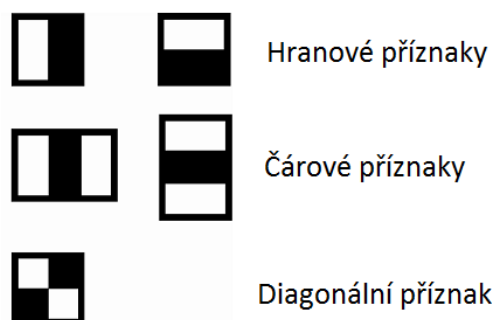
- Haarovy příznaky
Slouží k získání množiny příznaků, ze kterých vybírá AdaBoost algoritmus.
- Integrální obraz
Pomocí převodu klasického obrazu na obraz integrální zásadně zrychlíme výpočet Haarových příznaků.
- AdaBoost algoritmus
Jedná se o klasifikační algoritmus vycházející ze strojového učení, který iteračně vybírá vhodné příznaky pro detekci.

Detektor je vytvořen pro základní velikost podobrazu, kterým se následně prochází celý obraz a hledá kde detekuje objekt. Aby byl detektor nezávislý na velikosti

objektu, je po projití celého obrazu, podobraz zvětšen a opět se prohledává celý vstupní obraz.

3.6.1 Haarovy příznaky

Detektor Viola-Jones je založen na kooperaci mnoha jednoduchých příznaků s minimálními výpočetními nároky. Tyto příznaky jsou založeny na Haarových vlnkách. Haarovy příznaky jsou založeny na rozdílu sum hodnot jasů pixelů v jednotlivých oblastech. Základní příznaky používané detektorem jsou na Obrázek 11.



Obrázek 11: Haarovy příznaky

Jednotlivé příznaky jsou použity jak pro všechny pozice příznaku v podobrazu, tak i pro všechny velikosti příznaku které se vejdu do podobrazu.

V případě potřeby můžeme definovat nepřeberné množství dalších i složitějších příznaků. Ovšem i těchto 5 základních nabízí pro podobraz o velikosti 24x24 pixelů 180 000 různých příznaků.

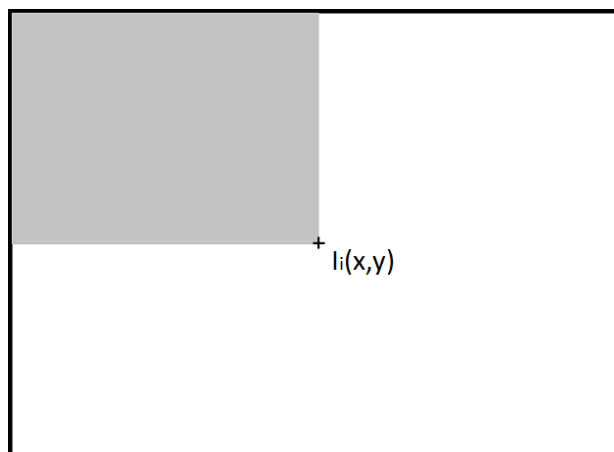
Odezva příznaku je dána sumou hodnot pixelů v bílé oblasti, od které je odečtena suma hodnot pixelů náležících do černé oblasti.

3.6.2 Integrální obraz

Výpočet odezvy Haarových příznaků obsahuje opakující součet pixelů pro různé oblasti. Tyto výpočty jsou výpočetně náročné a vzhledem k velkému počtu příznaků by tím celý detektor přišel o základní vlastnost rychlého výpočtu. Tento problém však elegantně řeší integrální obraz. Tento obraz se spočítá pouze jednou pro a poté se použije pro výpočet všech příznaků. Integrální obraz I_i odpovídá součtu všech předcházejících bodů v obraze dle rovnice. Princip integrálního obrazu je lépe patrný z následujícího obrázku.

$$s(x, y) = s(x, y - 1) + I(x, y) \quad (3-1)$$

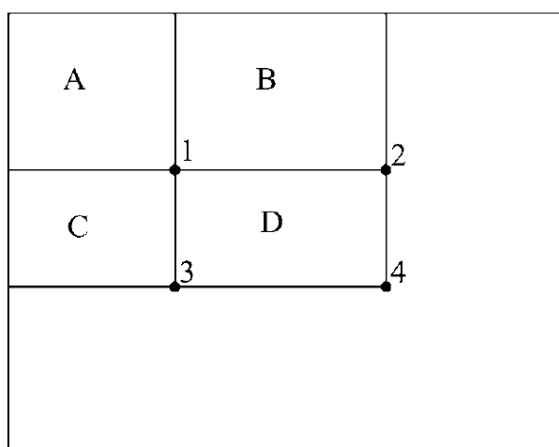
$$I_i(x, y) = I_i(x - 1, y) + s(x, y) \quad (3-2)$$



Obrázek 12: Integrální obraz. Hodnota bodu $I_i(x, y)$ se rovná sumě pixelů původního obrazu v šedém obdélníku.

Výpočet sumy pro jakoukoliv obdélníkovou oblast v obraze je dán pouze dvěma operacemi sčítání a jednou operací odčítání. Čímž se výpočet odezvy příznaku stává velmi efektivním. Výpočet sumy obdélníku D z následujícího obrázku je roven

$$\text{suma(D)} = I_i(4) + I_i(1) - [I_i(2) + I_i(3)]. \quad (3-3)$$



Obrázek 13: Příklad výpočtu sumy pixelů z integrálního obrazu.

3.6.3 AdaBoost

AdaBoost, zkratka Adaptive Boosting, patří mezi algoritmy strojového učení, které slouží ke zlepšení přesnosti klasifikátorů. Algoritmus je založen na vytvoření silného klasifikátoru, který kombinuje více slabých klasifikátorů.

Slabý klasifikátor je takový klasifikátor, který má úspěšnost jen o málo lepší než náhodný klasifikátor (úspěšnost přes 50%). Detektor Viola-Jones používá slabé

klasifikátory založené na výše popsaných Haarových příznacích, jejichž výstup je prahován dle naučeného prahu, individuálního pro každý příznak.

Algoritmus AdaBoost vybírá z množiny slabých klasifikátorů dle jejich úspěšnosti na trénovací množině. Trénovací množina je váhována váhami w_t . Ty jsou inicializovány rovnoměrně na celou množinu. Algoritmus pak pracuje dle následujících kroků.

1. Výběr klasifikátoru s nejmenší chybou klasifikace
2. ověření, že chyba klasifikátoru není větší než 0,5
3. aktualizace vah trénovacích vzorů, dle toho zda byl vzor klasifikován správně, či špatně.

Pokud byl klasifikován správně, hodnota váhy se nemění. Pro špatně klasifikované vzory se váha zvětší a v další iteraci má tento vzor větší vliv na výsledek klasifikace. To způsobí, že další vybraný klasifikátor lépe klasifikuje v předešlém kroku špatně detekované vzory.

Výsledný silný klasifikátor, který spojuje všechny slabé klasifikátory do jednoho se nazývá perceptron. Vstupem do perceptronu jsou binární výstupy slabých klasifikátorů, které jsou váhovány dle jejich úspěšnosti klasifikace.

3.6.4 Kaskáda klasifikátorů

I přes rychlý výpočet jednotlivých příznaků by byl výsledný algoritmus kvůli velkému počtu klasifikátorů pomalý. Tento problém je vyřešen použitím kaskádního řazení klasifikátorů. Podobraz je postupně klasifikován vytvořenými klasifikátory. Pokud ho některý stupeň kaskády klasifikuje negativně, je z klasifikace definitivně vyřazen a nemusí projít všemi klasifikátory. Tato vlastnost byla odvozena od skutečnosti, že většina oken v obraze je negativních. Tím dokážeme znatelně zrychlit detekci v obraze. První klasifikátor v kaskádě je postaven tak, aby odmítl co nejvíce negativních podoken, ale aby propustil všechny pozitivní. Další stupně kaskády mají za úkol stejný cíl s tím rozdílem, že mají těžší úlohu v potřebě přesnější detekce obličeje. Avšak jejich složitost se může postupně zvyšovat, protože se již nepoužijí pro klasifikaci všech podoken. V ideálním případě by všemi stupni kaskády měly projít všechny pozitivní podokna a negativní podokna by měli být vyřazeny z kaskády co nejdříve.

4 OPTICKÉ MĚŘENÍ POZICE VE 3D

Snímáním scény kamerou transformujeme 3D prostor na 2D obraz. Tím je ztracena informace o hloubce. Tato kapitola je věnována rekonstrukci ztraceného rozměru z obrazu.

4.1 Přehled metod pro optické měření pozice ve 3D

Zpětné získání třetího rozměru z obrazu je možné dosáhnout použitím upravených zařízení společně se zpracováním obrazu. Existuje více metod, které vyžadují různé úpravy hardwaru, či složitější algoritmy vyhodnocení.

Hlavní principy optického měření pozice ve trojrozměrném prostoru[8.]:

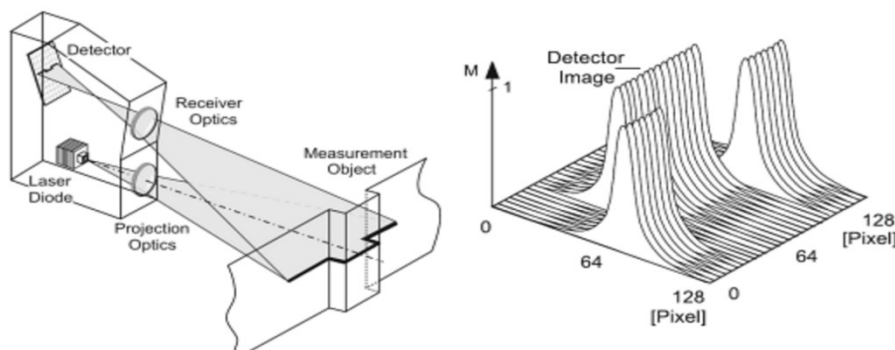
- Triangulace – aktivní, pasivní, měření s teodolitem
- Podoba ze stínování
- Optická interferometrie
- Měření doby letu modulovaného signálu
- Fokusovací techniky

Triangulace je nejvíce rozšířená technika v počítačovém vidění. Výrazné zlepšení aktivní triangulace, pasivní triangulace a kombinace obou metod přináší velký pokrok směrem k 3-D triangulační kameře a stereovizi v reálném čase.

4.1.1 Aktivní triangulace

Aktivní triangulaci tvoří na jedné straně zdroj strukturovaného světla a na druhé straně senzor. Zdroj světla může být jediný paprsek, který na měřeném objektu vytváří světelný bod. Poté mluvíme o 1-D laserové triangulaci. Úhel na straně zdroje světla α je neměnný a úhel na straně detektoru β se mění v závislosti na vzdálenosti objektu. Tento úhel je měřen fotocitlivým detektorem, kterým je CCD snímač. Z úhlu můžeme určit neznámou hloubku z (vzdálenost mezi senzorem a objektem). K výpočtu ještě potřebujeme znát neměnnou velikost báze senzoru b .

Pokud je jako zdroj světla použit světelný pruh mluvíme o 2-D triangulaci.

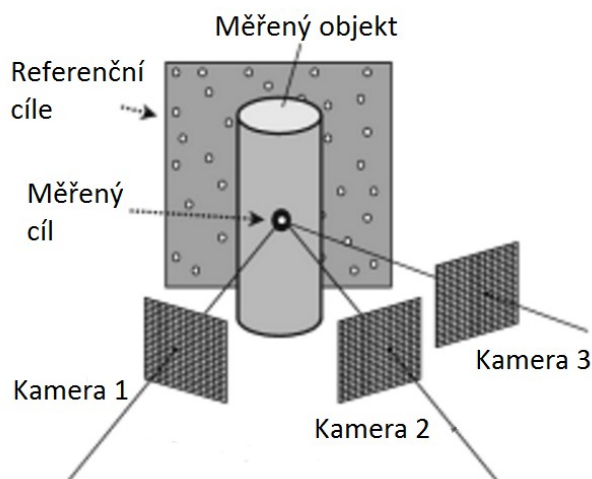


Obrázek 14: Příklad 2-D aktivní triangulace, [8.]

4.1.2 Pasivní triangulace

Pasivní v tomto kontextu znamená, že není uvažováno geometrické uspořádání zdroje světla. Metoda pasivní triangulace vyžaduje dva snímky stejné scény z rozdílných pozic. Toho lze docílit pomocí dvou kamer, u kterých známe přesnou vzájemnou pozici. Pro statickou scénu lze použít i pouze jedna kamera, kdy se nejprve nasnímá obraz pro jednu pozici, následně se pozice změní a poté se opět vyfotí scéna. Požití dvou kamer s paralelními optickými osami se nazývá stereovize (podrobněji popsána níže).

V získaných obrazech je potřeba nalézt korespondenční body. Z jejich obrazových souřadnic na snímcích lze následně vypočítat pozici objektu ve 3-D. K tomu však musíme znát geometrické uspořádání kamer, anebo použít samokalibrující se metody použitím referenčního cíle. Tím získáme metodu, která je odolná vůči nechtěnému posunutí polohy kamer, ať již vibracemi, či opotřebením. Viz Obrázek 15.



Obrázek 15: Příklad samokalibrující pasivní triangulace. Převzato z [8.]

4.1.3 Měřicí systém s teodolitem

Tato metoda patří mezi nejpresnější triangulační metody schopny měřit s relativní přesností okolo 10^{-6} . Avšak nevýhodou je dlouhá doba měření. Princip spočívá v zaměření alespoň dvou teodolitů na měřený cíl. Úhly natočení jsou měřeny elektronicky. 3D souřadnice jsou určeny z naměřených úhlů a známých pozic teodolitů.

4.1.4 Podoba ze stínování

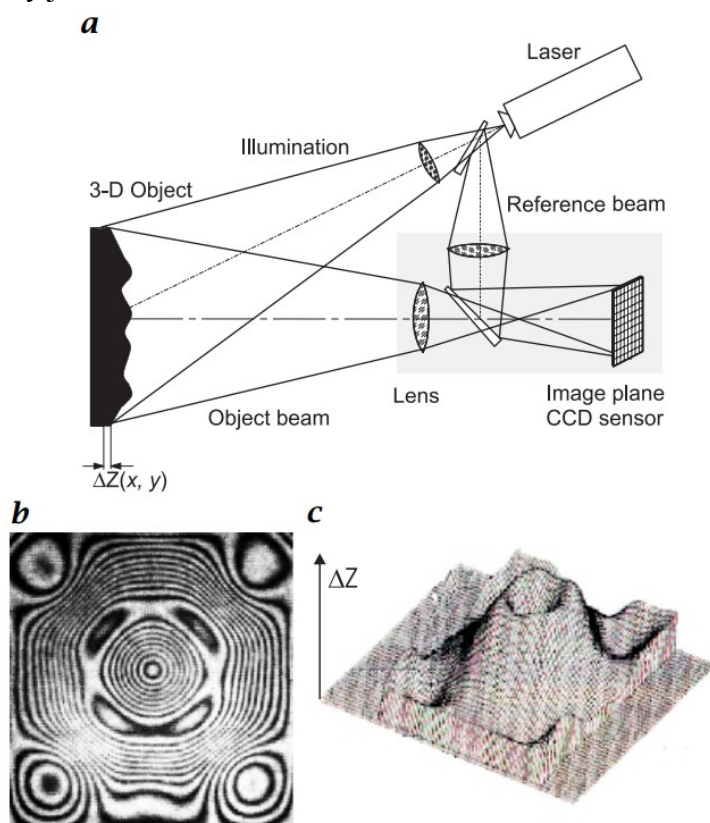
Algoritmus je založen na známé pozici kamery a zdrojů světla a z ozáření, stínů a odlesků. Z těchto údajů je možno určit normály povrchových elementů. Používají se rozdílné techniky zahrnující použití více snímků s rozdílným osvětlením a nebo sekvenci snímků s pohybujícím se osvětlením.

4.1.5 Optická interferometrie

Optická interferometrie je založena na interferenci odražené a referenční vlně. Změna vzdálenosti cílového objektu odpovídá fázovému posuvu interferované vlny. Pomocí tohoto principu nelze měřit absolutní vzdálenost. Jednoznačně lze určit vzdálenost jen v rozsahu $\lambda/2$ použitého světla. Velká přesnost měření vzdálenosti ovšem vyžaduje velmi stabilní podmínky měření a nebo použití dvou interferometrů ke kalibraci. Z toho plyne nevýhoda optické interferometrie, kterou je vysoká cena senzoru, kvůli složitému a preciznímu zařízení.

Mezi nejpoužívanější interferometry patří: Michelsonův, Sagnacův, Fabry-Peretův a další. Tyto zařízení se liší především celkovým uspořádáním optických prvků a způsobem rozdělení paprsků.

Zajímavou možností využití optické interferometrie v počítačovém vidění je měření povrchu objektu. Interferencí odraženého paprsku od nerovností povrchu společně s referenčním paprskem vzniká tzv. skvrnový interferogram. Vzdálenost mezi jednotlivými pruhy je $\lambda/2$.



Obrázek 16: Využití interferometru pro měření povrchu objektu. a) principiální schéma snímače, b) skvrnový interferogram, c) výsledný model změřeného povrchu (vyžaduje alespoň 3 měření s rozdílnou vzdáleností senzoru od objektu). [8.]

4.1.6 Měření doby letu modulovaného signálu

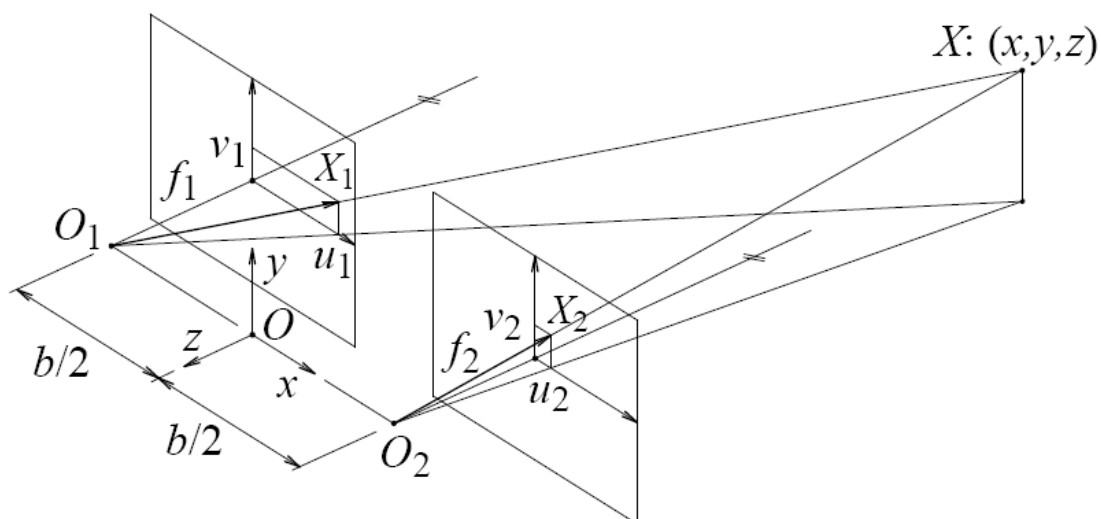
Vzdálenost k objektu lze jednoduše měřit pomocí doby mezi vyslaným a přijatým signálem (TOF – Time Of Flight), který se odrazil od objektu. Vzdaľenost z se spočítá z doby letu τ a rychlosti světla c .

$$z = \frac{c \cdot \tau}{2} \quad (4-1)$$

Veliká rychlost světla, přibližně $300 \text{ m}/\mu\text{s}$, klade nároky na rychlost vyhodnocovací jednotky. Další nevýhodou je mrtvá doba, kdy senzor není schopen měřit a nižší přesnost.

4.2 Stereovize

Stereovize, či stereovidění je metoda spadající mezi pasivní triangulační metody. Jedná se o metodu se dvěma kamerami, které jsou umístěny tak, aby měly optické osy paralelně. Viz Obrázek 17.



Obrázek 17: Stereovize – dvě kamery s paralelními optickými osami.[9.]

Pokud jsou splněny následující podmínky, dostáváme speciální a nejjednodušší případ stereovize; Souřadné osy u_1 a u_2 leží na přímce, počátek globálního souřadného systému dělí vzdálenost optických středů b na polovinu a osy globálního souřadného systému jsou rovnoběžné s osami kamer $x \approx u_1(u_2)$ $y \approx v_1(v_2)$ a ohniskové vzdálenosti obou kamer jsou stejné $f=f_1=f_2$.

Uvažujme ve scéně bod $X(x, y, z)$. Tento bod se promítá do obrazové roviny kamer jako bod $X_1(u_1, v_1)$ pro první kameru a $X_2(u_2, v_2)$ pro druhou kameru.

Z podobnosti trojúhelníků vychází vztahy jejichž řešením získáme pro souřadnice bodu následující rovnice [9.]:

$$z = -\frac{\mathbf{b} \cdot \mathbf{f}}{\mathbf{u}_1 - \mathbf{u}_2} \quad (4-2)$$

$$x = \frac{\mathbf{u}_1}{\mathbf{u}_1 - \mathbf{u}_2} \quad (4-3)$$

$$y = \frac{\mathbf{v}_1}{\mathbf{u}_1 - \mathbf{u}_2} \quad (4-4)$$

Proměnná $f(px)$ v tomto případě udává zobrazovací konstantu kamery.

4.2.1 Korespondenční problém

Problémem u řešení algoritmu stereovize je automatické nalezení bodu z prvního obrazu ve druhém. Pokud by nebyl bod jednoznačně definovaný (např. značka na objektu) a tudíž ho nešlo v obraze s jistotou identifikovat, bylo by řešení velmi komplikované. Avšak tuto komplikaci pomáhá vyřešit tzv. epipolára. Epipolára je dána spojnicí ohniska kamery, nalezeného bodu ve snímku a bodu na objektu, která je promítnuta do roviny druhé kamery. Hledaný bod ve druhém obraze leží na této epipoláře. Nalézt ho lze pomocí korelace okolí bodu s body na epipoláře a nalézt kde dosahuje korelační funkce maximum. Úskalím je vhodná volba velikosti okolí bodu. Pokud by okolí bylo velmi malé, mohlo by dojít k falešným detekcím. Pokud však okolí bude zas příliš velké, úplatní se zde skutečnost, že oba obrazy nejsou úplně identické, protože jsou pořízeny z různého místa. V reálné obecné scéně se musíme smířit s faktem, že mohou nastat případy, kdy nelze jednoznačnost zaručit. [9.]

5 ROZBOR ÚLOHY

Automatickým natáčením zpětného zrcátka v závislosti na poloze očí řidiče je myšleno takové natočení zrcátka, aby řidič v jakékoliv pozici hlavy vůči zrcátku viděl v zrcátku definovaný prostor za autem.

V zadání není upřesněno, má-li se jednat o vnitřní zpětné anebo boční zpětné zrcátko automobilu. Pokud se bude jednat o vnitřní zpětné zrcátko, stačí použít jednu kameru, která bude umístěna co nejblíže zrcátku (např. na stropě nad zrcátkem).

V případě bočních zrcátek by také bylo potřeba umístit kameru co nejblíže k ose zrcátko-očí řidiče. Poté by však bylo možné automaticky natáčet pouze jedno boční zrcátko. Důvodem je, že nelze pomocí kamery přesně rozpoznat přiblížení objektu ke kameře, pokud tento objekt nemá stálou velikost. Přiblížení ovšem způsobuje výrazný pohyb vůči druhému zrcátku. Řešením je použít jednu z metod 3D rekonstrukce obrazu. Výhodou bočních zrcátek je dostupná mechanika elektrického natáčení. To v případě vnitřního zrcátka bude muset být navrženo a zkonstruováno.

Vzhledem k výhodám a nevýhodám obou případů jsem si vybral řízení bočních zrcátek. V tomto případě je potřeba měřit polohu obličeje ve 3D a z výsledných dat je možné ovládat všechny tři zrcátka.

5.1 Popis scény

Ve scéně bude jako statické a téměř neměnné pozadí interiér automobilu. Světelné podmínky jsou ovšem dynamické. Detekce by měla spolehlivě fungovat při jasné obloze, za špatného počasí i v noci.

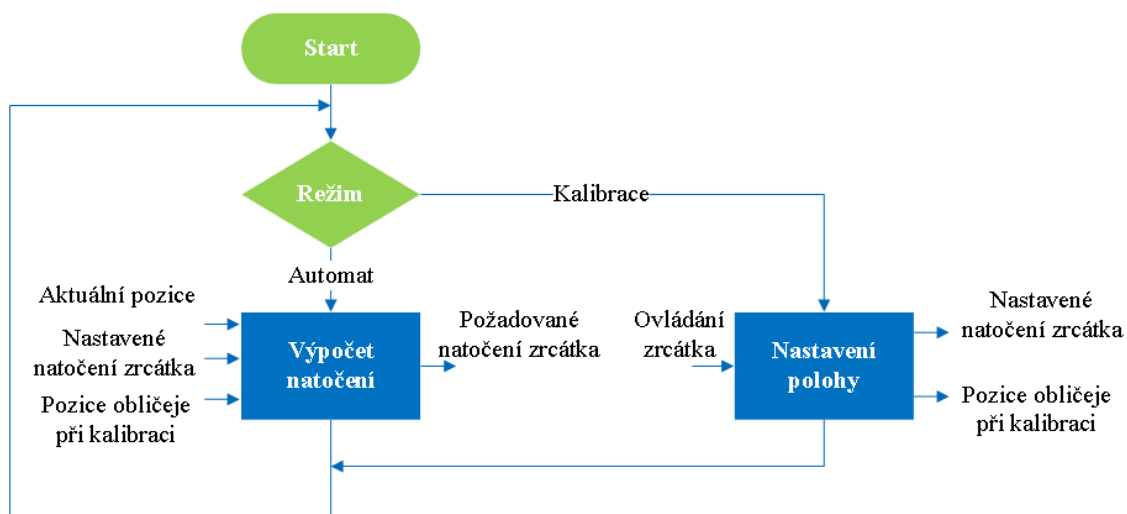
K zajištění tohoto požadavku je vhodné scénu nasvítit. Tento přísvit ale nesmí přímo oslňovat řidiče. Vzhledem k předpokládanému umístění kamer na palubové desce, nelze použít zdroj světla ve viditelném spektru. Nejvhodnější je volba infračerveného světla. Tím jsme schopni i bez oslnění a při nízké intenzitě zdroje nasnímat požadovanou scénu.



Obrázek 18: Použití IR přísvitu ve tmě v interiéru automobilu.

5.2 Funkce zařízení

Zařízení má za úkol sledovat polohu obličeje a dle ní měnit natočení zrcátka tak, aby byl v zrcátku stále vidět definovaný prostor. Definovaný prostor vyjadřuje takový prostor, kam se zrcátko nastaví při takzvané kalibraci. Kalibraci lze spustit přepnutím ovládání zrcátek z automatického režimu na nastavení levého, respektive pravého, zrcátka. V tomto režimu lze nastavit polohu zrcátka ovládáním tak, aby se v zrcátku objevil požadovaný prostor. K následným výpočtům zařízení potřebuje znát i polohu obličeje ve chvíli nastavování. Viz kapitola 7.9. Při přepnutí zpět do automatického režimu již zrcátko sleduje polohu obličeje.

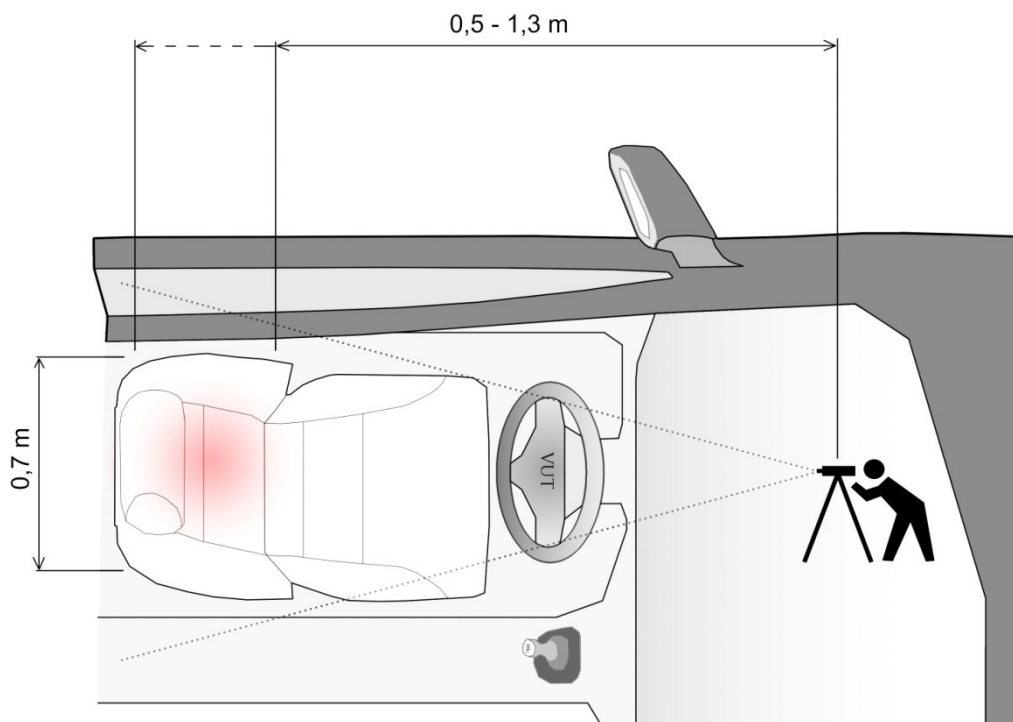


Obrázek 19: Diagram znázorňující základní princip zařízení

Nároky na rychlost detekce polohy nejsou veliké. V této aplikaci postačí, pokud zařízení stihne reagovat na změnu obličeje každou sekundu.

5.3 Měření polohy obličeje

Na následujícím obrázku je popsán prostor řidiče, dle kterého lze určit rozsah vzdáleností, ve kterém musí zařízení měřit polohu. Vzdálenost obličeje od kamery s uvažovanou rezervou pro různé typy aut se pohybuje od 0,5 do 1,3 metru. Rozsah do stran je uvažován v rozmezí 0,8 metru. Pozice hlavy ve vertikálním směru záleží na velikosti řidiče a nastavení polohy sedadla. I s rezervou se tato pozice nachází v rozmezí 0,5 metru.



Obrázek 20: Pozice hlavy řidiče v automobilu

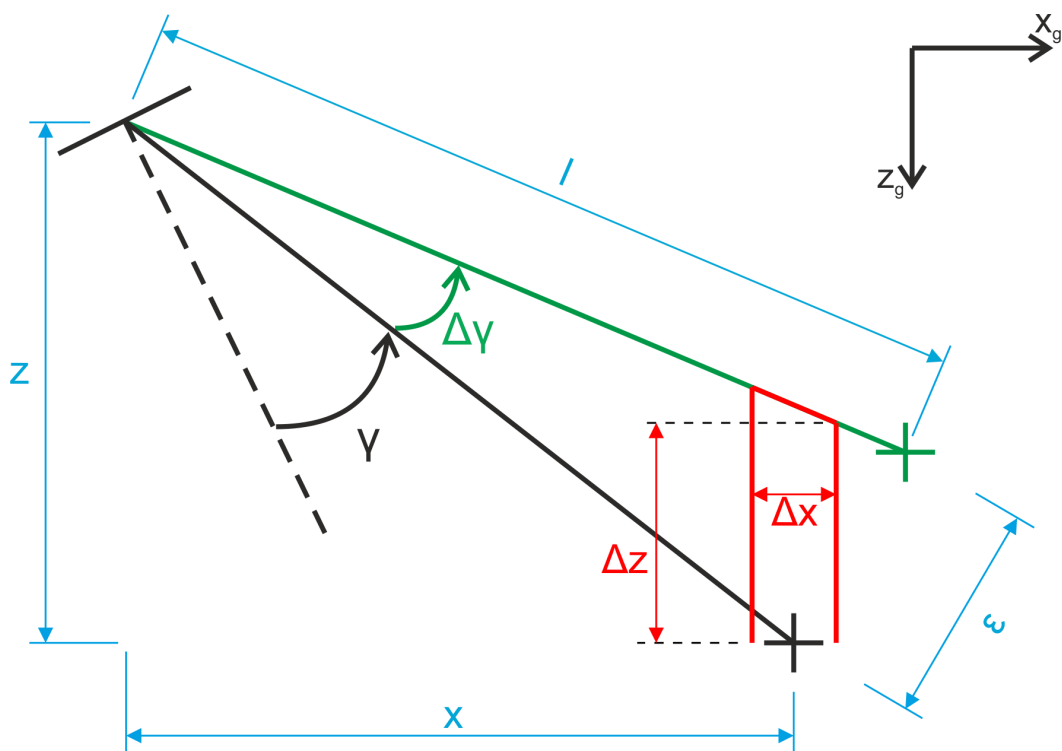
Z metod optického měření 3D pozice připadají v potaz pouze měření, která zvládnout měřit v požadovaném rozsahu. Z výčtu v kapitole 4 lze vybrat z aktivní triangulace, pasivní triangulace a měření doby letu modulovaného signálu.

Vzhledem k možnému pohybu objektu a k jednoduchosti implementace byla vybrána metoda stereovidění spadající do pasivní triangulace.

K použití stereovidění jsou potřeba dvě kamery. Tím se zvyšují nároky na výpočetní jednotku. Avšak k implementaci postačí provést stejný postup lokalizace tváře u obou snímků a z výsledných souřadnic jednoduše spočítat pozici obličeje vůči kameře (kapitola 4.2).

5.3.1 Citlivostní analýza stereovize

Výpočet velikosti báze stereovize je vhodné začít od konce. To znamená nejprve určit, jakou chybu lze tolerovat ve výhledu v zrcátku. Chyba byla zvolena $\varepsilon_0=5$ cm ve výhledu na úrovni zadního rohu automobilu ($l_0=4$ m pro běžný osobní automobil). Tuto chybu je potřeba přepočítat na stranu řidiče.



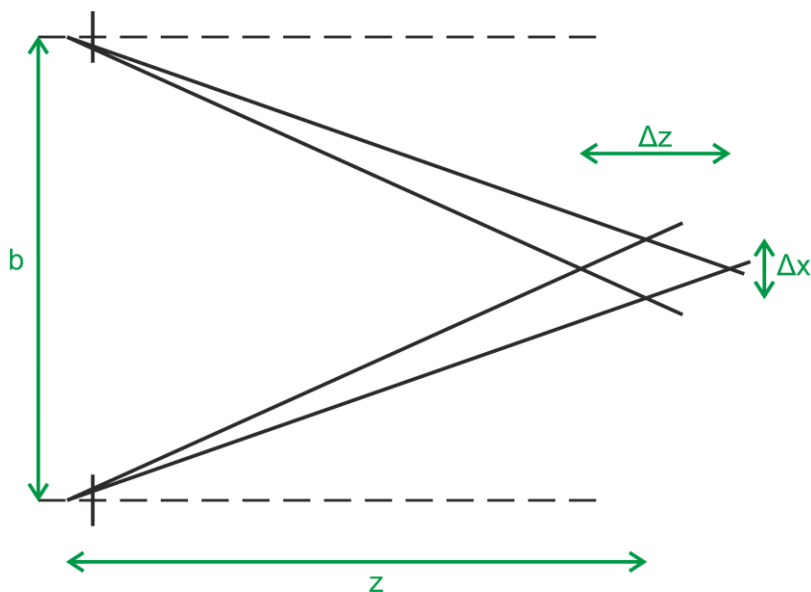
Obrázek 21: Znázornění přepočítané chyby ε polohy řidiče ze známých vzdáleností l_0 a l . Červeně je vyznačen přepočet chyby do souřadnic x a z .

Hodnotu chyby ε vypočítáme dle vztahu

$$\varepsilon = \varepsilon_0 \frac{l}{l_0} \quad (5-1)$$

kde ε_0 je nadefinovaná chyba ve vzdálenosti l_0 a l je vzdálenost obličejce od zrcátka.

Pro vyhodnocení citlivosti stereovize, je chyba přepočítána do z -souřadnice, při uvažování chyby Δx rovnu záběru jednoho pixelu ve vzdálenosti od kamery z_0 . K dalším výpočtům je nutné znát parametry kamery, konkrétně použité rozlišení a pozorovací úhel kamery.



Obrázek 22: Citlivost stereovize

$$\Delta x = \frac{2 * z_0 * \tan\left(\frac{\varphi}{2}\right)}{n_h} \quad (5-2)$$

Kde φ označuje pozorovací úhel kamery a n_h počet pixelů v horizontální rovině.

Změněná souřadnice x_2 je rovna

$$x_2 = x_1 + \frac{\Delta x}{2} \quad (5-3)$$

Dále je nutné znát změnu úhlu $\Delta\gamma$.

$$\gamma = \tan^{-1}\left(\frac{x_1}{z_1}\right) \quad (5-4)$$

$$\Delta\gamma = 2 * \tan^{-1}\left(\frac{\epsilon/2}{1}\right) \quad (5-5)$$

$$\gamma_2 = \gamma_1 + \Delta\gamma \quad (5-6)$$

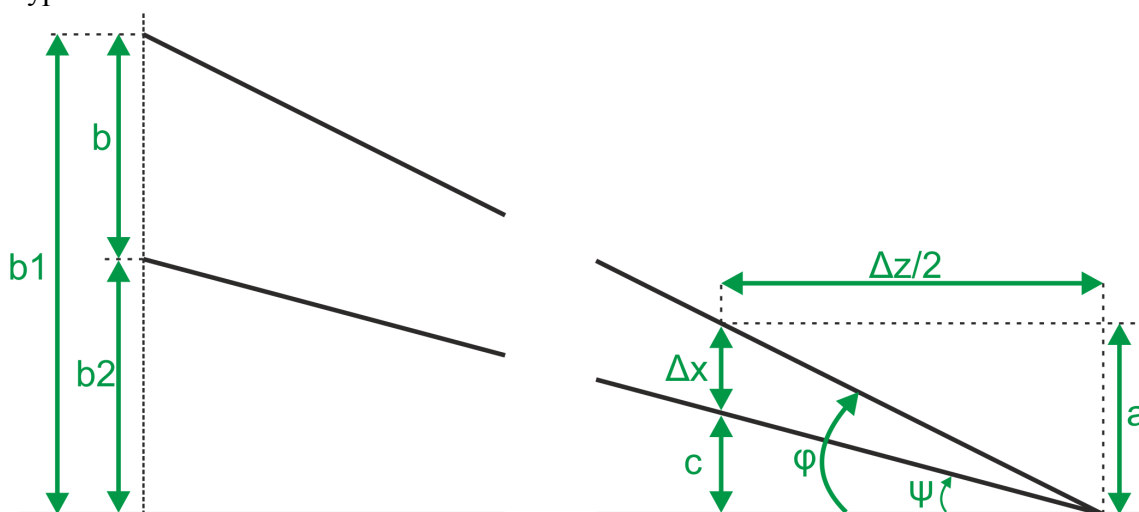
Z těchto hodnot už lze vypočítat maximální povolenou chybu v ose.

$$z_2 = \frac{x_2}{\tan \gamma_2} \quad (5-7)$$

$$\Delta z = z_1 - z_2 \quad (5-8)$$

Pro oba definované případy je počítáno s tím, že se obličej v daných souřadnicích nachází na kraji výseče pravé kamery. V tomto případě, bude chyba Δz největší pro danou vzdálenost od kamer.

Výpočet velikosti báze lze odvodit z obrázku 23 .



Obrázek 23: Geometrie výpočtu velikosti báze stereovize

Vztahy pro výpočet jsou:

$$a = \frac{\Delta z}{2} * \tan\left(\frac{\varphi}{2}\right) \quad (5-9)$$

$$c = a - \Delta x \quad (5-10)$$

$$\psi = \tan^{-1}\left(\frac{c}{\frac{\Delta z}{2}}\right) \quad (5-11)$$

$$b_1 = z * \tan\left(\frac{\varphi}{2}\right) \quad (5-12)$$

$$b_2 = z * \tan \psi \quad (5-13)$$

$$b = b_1 - b_2 \quad (5-14)$$

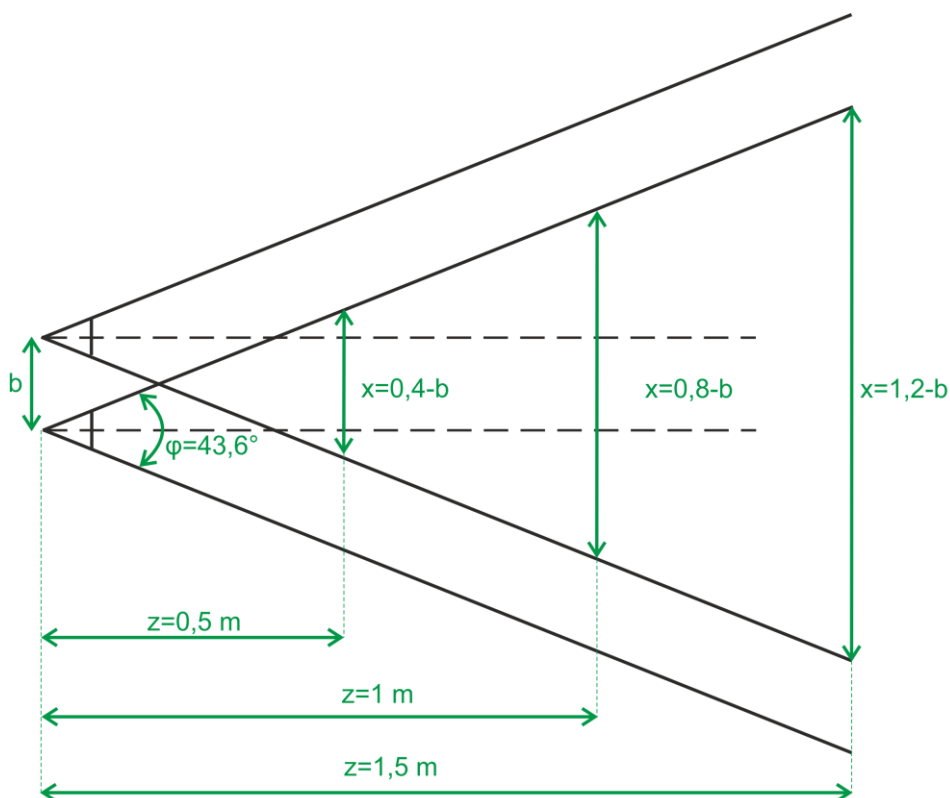
Pro výpočet byly zvoleny dvě limitní situace:

- První situace odpovídá nejbližší možné vzdálenosti obličejů od zrcátka:
O₁(x=0,4 m, z=0,4 m) Této situaci odpovídá l=0,56 m, ε=7,1 mm a Δz=9,2 mm.
- Druhým případem je největší vzdálenost obličejů od zrcátka: O₂(x=0,4 m, z=1,3 m). Pro tyto hodnoty byly vypočítány l=1,36 m, ε=17 mm a Δz=49,8 mm.

Umístění zrcátka vůči kameře bylo zvoleno dle maximálního možného posuvu v ose z v automobilu: M(x, z=0,2 m). Vybraná kamera, která bude popsána v kapitole 6.1, má rozlišení 320 x 240 px a pozorovací úhel φ=43,6°.

Pro první případ vychází velikost báze $b=24$ cm, pro případ druhý $b=22,6$ cm. Větší z vypočtených hodnot se použije pro definici minimální báze stereovize, aby byly zachovány požadavky na maximální chybu. Pro navrhované zařízení bude použita velikost báze $b=25$ cm.

Velikost báze má zásadní vliv na prostor, ve kterém lze pozici měřit. Čím větší bude velikost báze, tím se daná výšeč zužuje. Na následujícím obrázku lze vidět výšeč, ve které je použitá stereovize schopna měřit polohu objektu.



Obrázek 24: Vliv velikosti báze stereovize na výslednou výšeč pro vybrané kamery

6 NÁVRH HARDWARE

Cílem této kapitoly je výběr a popis HW vybavení; především kamery, výpočetní jednotky a elektrického polohování zrcátka.

6.1 Kamera

Prvním členem v řetězci zpracování obrazu je snímání. To zahrnuje kameru a osvětlení. Pro kameru v tomto případě nejsou kladeny vysoké nároky. Je uvažován reálný pohyb člověka ve scéně, proto rychlost snímání není tolik důležitá a požadavky splňují téměř všechny webkamery. Výhodou těchto kamer je i malá cena, připojení pomocí USB a snadné vyčítání dat.

Vzhledem k definované scéně (v automobilu) a malé vzdálenosti, není požadováno velké rozlišení kamery. A to i s ohledem na výpočetní nároky při zpracování. Dostatečným a dostupným standardem je rozlišení VGA; 640x480 px (0,35Mpix).

Dalším kritériem je požadavek na přísvit. Na trhu lze sehnat takovéto kamery s integrovaným IR přísvicením. Po průzkumu trhu byla vybrána webkamera A4Tech PK-333E, která splňuje požadavky na parametry i cenu a má integrovaný IR přísvit.

Základní vlastnosti kamery udávané výrobcem [10.]:

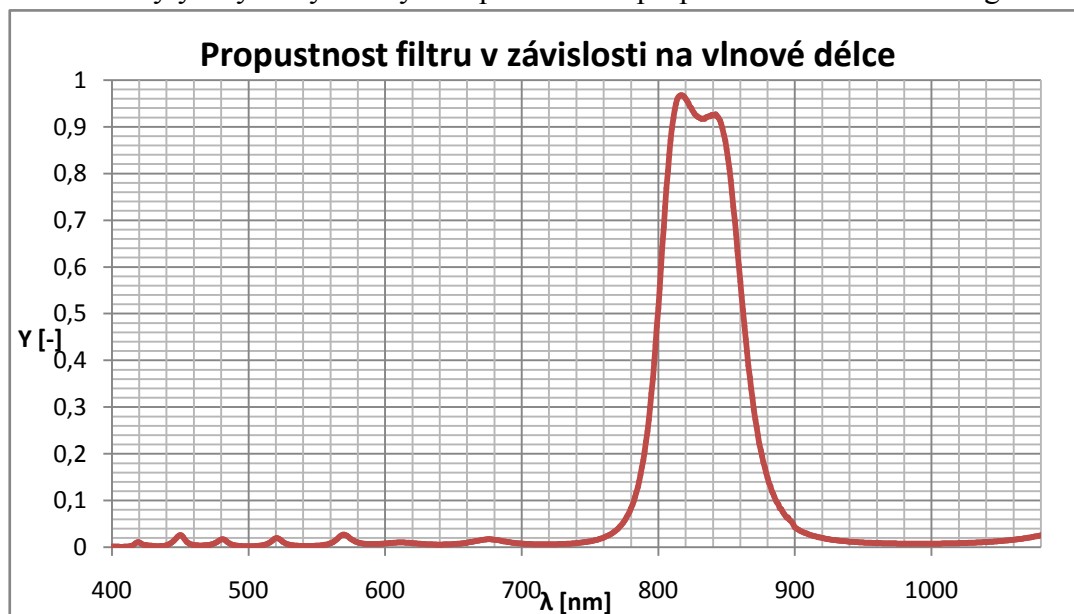
- Typ senzoru: CMOS 1/6" (2,4 x 1,8 mm)
- Rozlišení videa: VGA (640 x 480) při 30 sn./s
- Přísvicení: 6x infračervená LED
- Objektiv:
 - Ohnisko: $f=3$ mm
 - Světelnost: $F=2,4$
 - Pozorovací úhel: 33°
- Zaostřování: 10 cm až nekonečno
- Funkce:
 - Automatické ostření
 - Automatické vyvážení bílé
- Rozhraní: USB 2.0

S použitím přísvitu lze na snímcích rozpoznat obličej v denním světle i v noci. Pro sjednocení pořízených snímků při odlišných podmínkách byly před kameru přidány filtry.



Obrázek 25: Srovnání snímků pořízených s filtrem a bez filtru. Horní řada odpovídá snímkům bez filtru, dolní snímkům s filtrem. V levém sloupci jsou pak snímky pořízené v noci a v pravém při denním světle.

Změřená vyzařovací vlnová délka přísvitu kamery je $\lambda=825$ nm. Na tuto hodnotu byly vybrány filtry s pásmovou propustí zobrazenou v grafu níže.



Obrázek 26: Graf závislosti relativní propustnosti na vlnové délce světla

6.2 Zpětné zrcátko

Pro tento projekt bylo zvoleno boční zpětné zrcátko. Na trhu se dají pořídit takzvaná elektrická zrcátka s pamětí. U těchto zrcátek se dá nastavit poloha společně s polohou sedadla řidiče. A řídicí jednotka si toto nastavení zapamatuje. Tyto zrcátka obsahují senzory pozice zrcátka. Vzhledem k nutnosti znát polohu zrcátka v této aplikaci, je vhodné použít právě toto zrcátko. Odpadne tím konstrukční implementace jiného senzoru natočení do zrcátka.

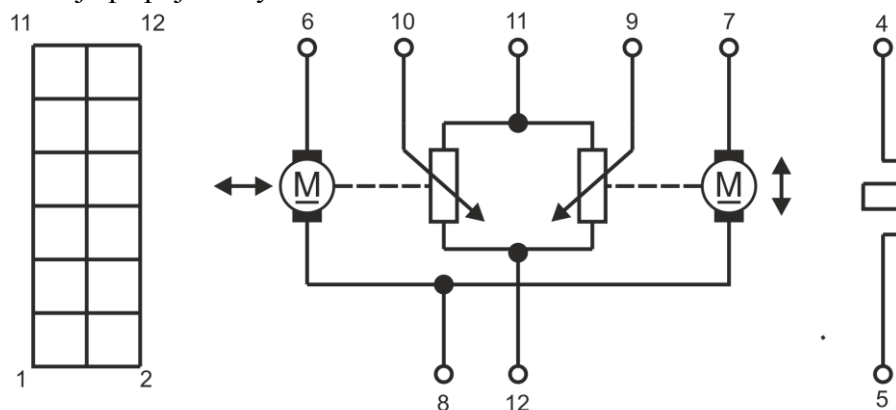


Obrázek 27: Fotka pořízeného elektrického zrcátka s pamětí pro vůz Škoda Octavia

Nakonec bylo vybráno elektricky ovládané zrcátko s pamětí pro osobní automobil Škoda Octavia III. Snímání natočení je zde řešeno velmi jednoduše pomocí dvou potenciometrů. Tyto potenciometry jsou konstrukčně zabudovány přímo do motorů.

V zrcátku jsou použity DC motory na 12V. Dále bude navrženo ovládání těchto motorů. Na motoru je převod na lineární pohyb, kterým je pohybováno se zrcátkem buďto směrem ven, nebo dovnitř.

Následující obrázek popisuje elektrické zapojení pořízeného zrcátka. Mezi kontakty 4 a 5 je připojeno vyhřívání zrcátka.



Obrázek 28: Schéma zapojení pořízeného zrcátka

6.2.1 Ovládání zrcátka

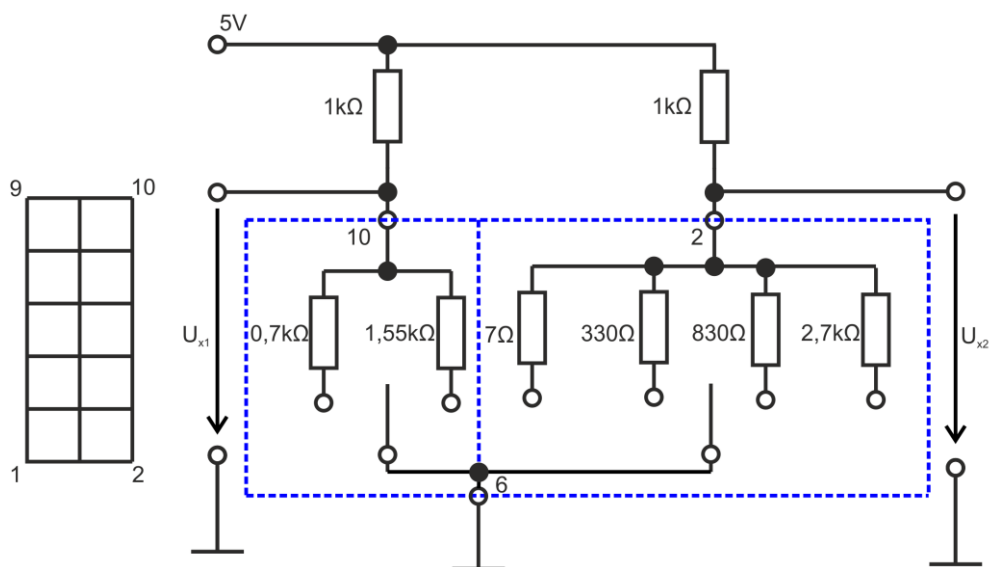
V automobilu se nastavuje zrcátko klasickým elektrickým ovládáním na dveřích řidiče. V projektu je použito stejné ovládání kvůli přiblížení se implementaci do automobilu. Opět byla vybrána součástka pro automobil Škoda Octavia III.



Obrázek 29: Použité ovládání zrcátek určené pro automobil Škoda Octavia

V tomto ovládání se využívá přepínání odporů o různých hodnotách. Viz následující schéma. K detekci přepnutí odporů lze využít odporový dělič. Změřená hodnota AD převodníkem bude rovna

$$U_x = 5 \frac{R_x}{1k + R_x}. \quad (6-1)$$



Obrázek 30: Schéma vnitřního zapojení ovladače zrcátek a připojení k vyhodnocovacímu obvodu

6.3 Výpočetní jednotka

Klíčový je výběr výpočetní jednotky, která bude schopna zpracovávat data ze dvou kamer s přijatelnou dobou zpracování.

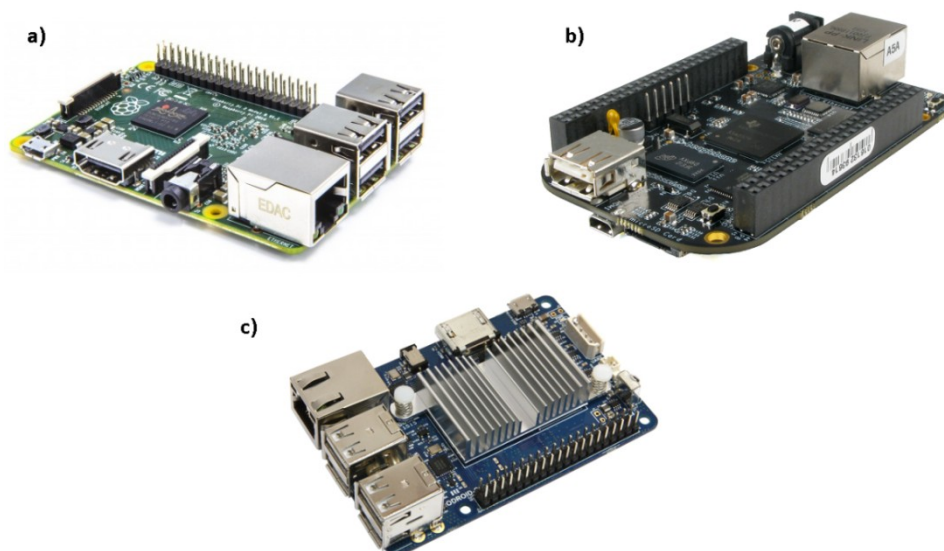
Na snímku z kamery je potřeba provést předzpracování, segmentaci a klasifikaci. Pro algoritmy bude vybrána vhodná výpočetní jednotka, která bude mít dostatečný výkon pro tyto úkony. A také bude mít vhodné rozhraní pro kamery a pro elektrické polohování zrcátka.

Základní požadavky na výpočetní jednotku:

- Dostatečný výkon pro zpracování snímků ze dvou kamer
- Periferie podporující připojení kamer – 2x USB
- Podpora připojení driverů pro motory, tlačítek pro ovládání a vstupů ze senzorů zrcátek
- Kompaktní rozměry
- Ideálně podpora knihovny obsahující zpracování obrazu (např. OpenCV)

Jednou z možných voleb, které přichází v úvahu, je Raspberry Pi 2. Tento modul disponuje potřebnými periferiemi pro kamery, dostatkem digitálních vstupů a výstupů. Jednotka je dodávána s operačním systémem Linux pod oficiální distribucí Raspbian. Použití Linuxu má výhodu v tom, že lze při programování aplikace využít knihovnu OpenCV a také není těžké vytvořit grafické rozhraní pro testování.

Na trhu jsou i další velmi podobné alternativy, jako například BeagleBone Black, nebo moduly Odroid. Tyto moduly mají srovnatelné vlastnosti a výpočetní výkon.



Obrázek 31: Vhodné embedded moduly. a) Raspberry Pi 2, b) BeagleBone Black, c) Odroid-C1+

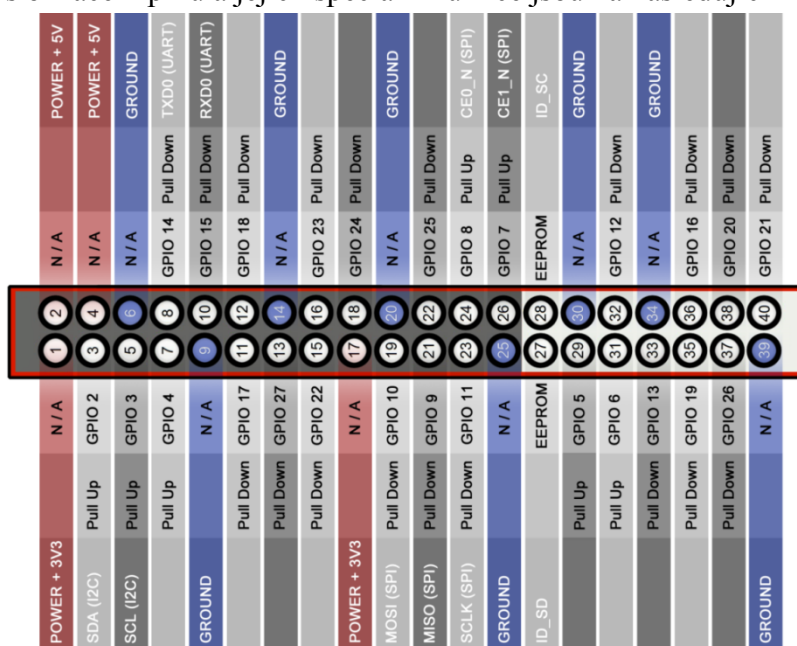
6.3.1 Raspberry Pi 2

Kvůli nejširší dostupné podpoře byla vybrána jednotka Raspberry Pi 2 model B. RPi 2 je jednodeskový počítač. Jednotka kromě dalšího obsahuje čtyř-jádrový procesor s frekvencí 900 MHz, 4 USB konektory, HDMI výstup a GPIO konektor (digitální vstupy/výstupy). Shrnutí parametrů je uvedeno v tabulce níže. RPi 2 jako paměť pro systém a data používá slot na mikro SD kartu. Z té je spouštěn operační systém, který lze volit z široké nabídky.

Tabulka 1: Hardwarové parametry Raspberry Pi 2

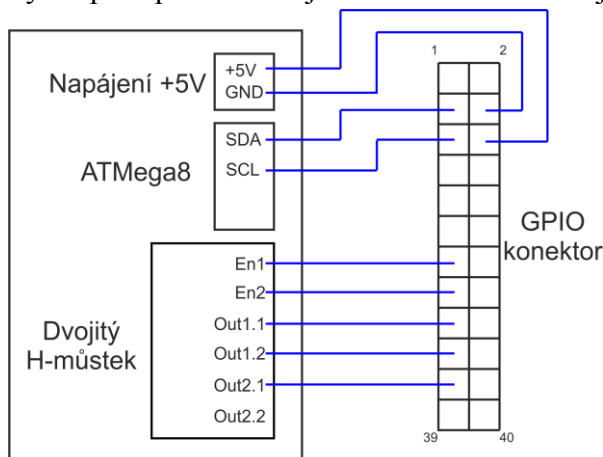
Model	RaspberryPi 2
CPU	BCM2836 ARM Cortex-A7, 4 jádra, 900 MHz
GPU	VideoCore IV
RAM	1GB SDRAM
Připojení k internetu	Ethernet RJ45 10/100 Mbit
USB	4x USB 2.0
Video výstup	HDMI
Audio výstup	3,5 mm jack, HDMI
GPIO konektor	40 pinů
Úložiště	micro SD karta
Rozměry	8,56 cm x 5,65 cm

GPIO konektor RPi 2 obsahuje celkem 40 pinů. Z toho 12 pinů slouží k distribuci napájení, 2 piny k identifikaci EEPROM a zbylých 26 pinů lze použít jako digitální vstupy a výstupy. Některé z pinů mají speciální funkce, které lze využívat. Především jako porty pro komunikaci (I2C, SPI, USART), nebo zabudované pull-up/pull-down odpory. Popis označení pinů a jejich speciální funkce jsou na následujícím obrázku.



Obrázek 32: Popis pinů a jejich funkcí GPIO konektoru Raspberry Pi 2

K řízení motorů, respektive H-můstků, bude použit GPIO konektor. Výběr pinů je třeba brát s ohledem na zabudované pull-up/pull-down rezistory. Ve vypnutém stavu Raspberry jsou na těchto pinech potenciály odpovídající jejich vnitřnímu zapojení. V praxi to znamená, že potřebujeme mít ve vypnutém stavu shodné potenciály pro ovládání H-můstku. V opačném případě jsou při vypnutí spuštěny motory zrcátka, což je ovšem nežádoucí. Výběr pinů pro zrcátka je zobrazen na následujícím schématu.

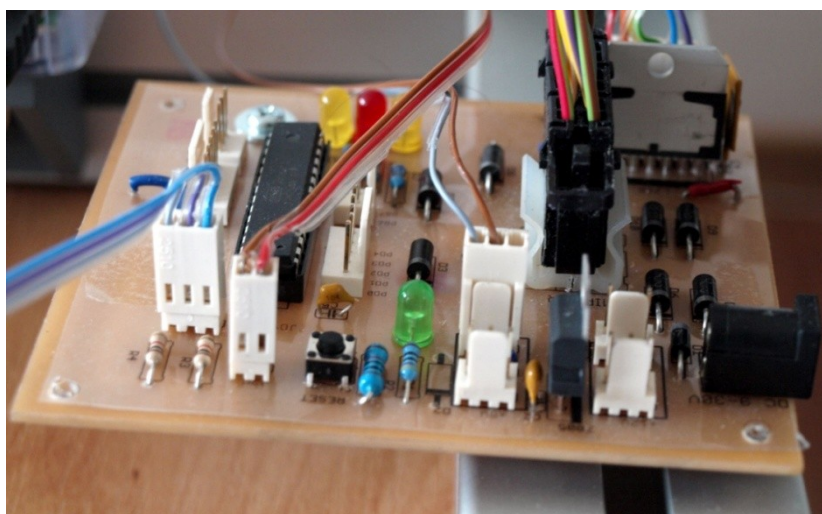


Obrázek 33: Znázornění použitých pinů konektoru GPIO a jejich zapojení

Jediné potřebné periferie, které tento počítač neobsahuje, jsou analogově-číslicové převodníky. Pro snímání polohy zrcátka a pro vyhodnocení přepínačů na ovládání je nutné k Raspberry připojit externí jednotku s AD převodníkem. Převodník může s Raspberry komunikovat po dostupných sběrnicích.

6.4 Rozšiřující deska

Pro potřebu externího AD převodníku, H-můstků pro řízení motorů a pro distribuci napájení byla navržena a vyrobena rozšiřující deska k Raspberry. Deska je z logického hlediska rozdělena do těchto třech celků.

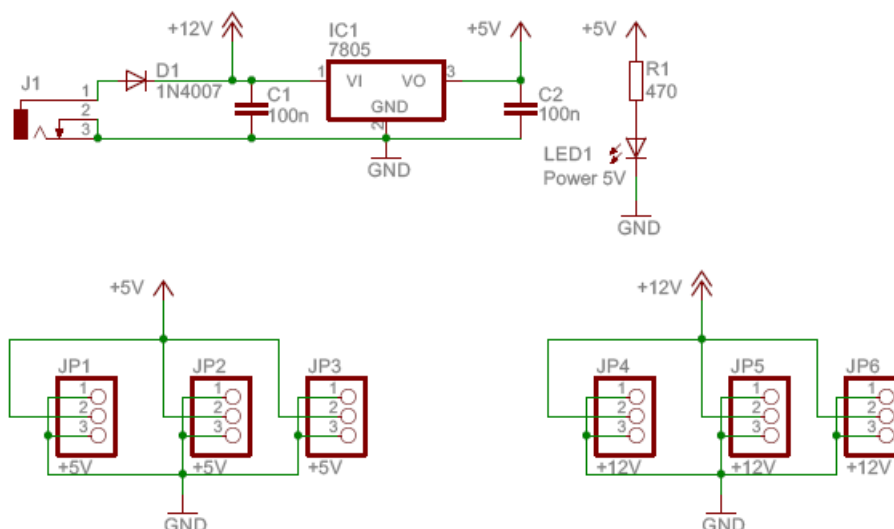


Obrázek 34: Vyrobená rozšiřující deska

6.4.1 Napájení

K napájení slouží běžný souosý konektor, ke kterému bude připojen 12V adaptér. Adaptér musí zvládnout dodat proud alespoň 2 A, aby pokryl proudový odběr RPi (650-900mA) , odběr motorů (2x 300 mA) a odběr ostatních částí.

Napětí pro motory je 12V. Další napájecí napětí je potřeba v úrovni +5V. K stabilizaci 5V napájení slouží regulátor napětí 78S05. Maximální výstupní proud regulátoru je roven $I_{omax}=2$ A. Přítomnost napájení je na desce značena diodou LED1.



Obrázek 35: Schéma napájecí části

Deska je zamýšlena jako testovací, a proto jsou na desce umístěny konektory pro rozvod napájení. Konektory JP1-JP3 pro napětí 5 V a JP4-JP6 pro napětí 12 V.

6.4.2 AD převodník

Pro potřeby zpracování analogového signálu byl použit 8bitový mikroprocesor ATmega8 od výrobce Atmel. Z důvodu přidané hodnoty, neboli možnosti předzpracování dat, byla upřednostněna ATmega8 před externími AD převodníky. Druhým důvodem ve výběru byla dostupnost tohoto mikroprocesoru. Základní parametry mikroprocesoru jsou uvedeny v tabulce.

Tabulka 2: Základní parametry mikrokontroléru ATmega8. Převzato z [12.].

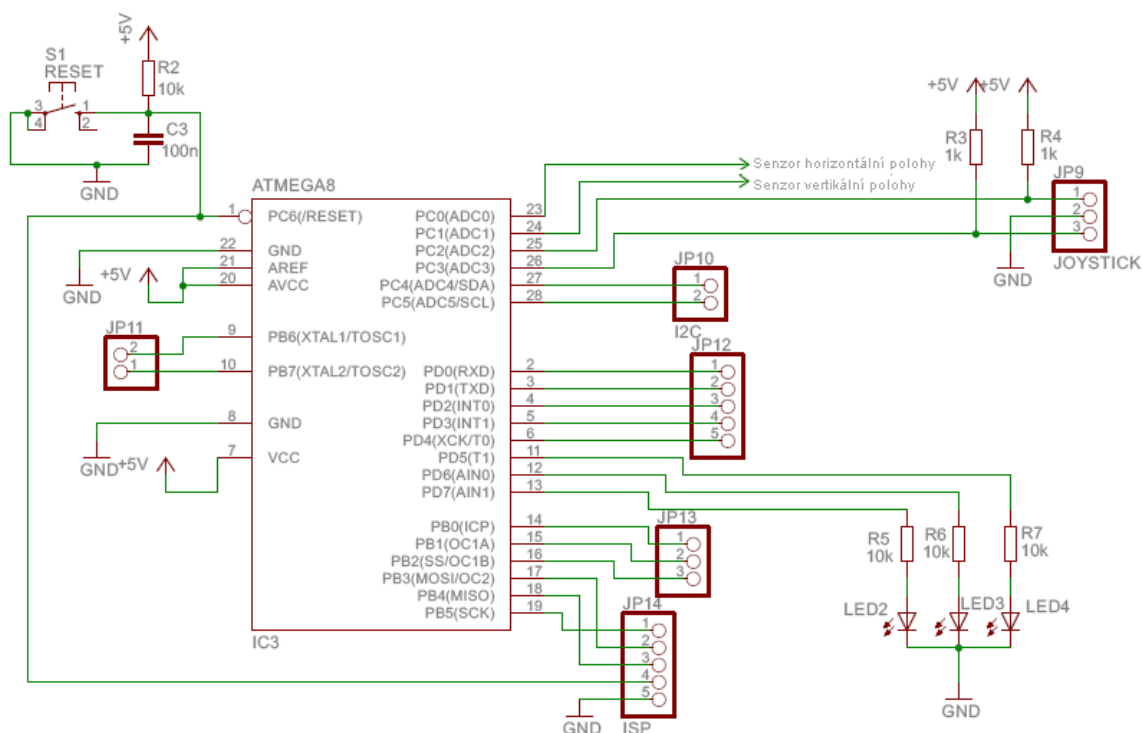
8bitová AVR RISC architektura
Paměť: 8kB flash, 1kB SRAM, 512B EEPROM
Interní oscilátor – 1, 2, 4, 8 MHz
23 programovatelných I/O pinů
2x8bitový + 1x16bitový čítač/časovač
3x PWM výstup
6x 10bitový A/D převodník
Komunikace: I2C, SPI, USART

ATMega8 má k dispozici vnitřní 8 MHz RC oscilátor. Tato frekvence procesoru s rezervou stačí na požadované operace. Čip obsahuje 6-ti kanálový AD převodník s podporovaným rozlišením 8, nebo 10 bitů. Převodník je typu s postupnou aproximací. V tomto projektu jsou využity čtyři kanály AD převodníku. Dva kanály pro senzory polohy zrcátka – konektor JP8, jeden kanál pro přepínač ovladače zrcátek a poslední pro vyhodnocení joysticku ovládání zrcátek – konektor JP9. Reference pro převodník je brána z napájecího napětí. Tím se zabrání vlivu zakolísání napájecího napětí na hodnotu z převodníku.

Ke komunikaci s Raspberry byla vybrána sériová komunikace I2C. Tato komunikace potřebuje pouze dva vodiče, datový vodič a hodiny. Komunikace je řízena master zařízením, který vysílá požadavky zařízením slave, dle jejich adresy. Komunikace je podporována na hardwarové úrovni oběma stranami.

K čipu jsou připojeny 3 LED diody k volitelné signalizaci stavu algoritmu. Na konektor JP14 jsou vyvedeny kontakty k tzv. ISP – In-System Programming. Přes tento konektor lze programovat procesor i bez jeho vyjmutí z desky. Tato skutečnost usnadňuje vývoj zařízení.

Ostatní nevyužité vstupně/výstupní piny jsou vyvedeny na konektory pro další případné použití. Zapojení je lépe patrné ze schématu.



Obrázek 36: Schéma zapojení ATMega8 na rozšiřující desce

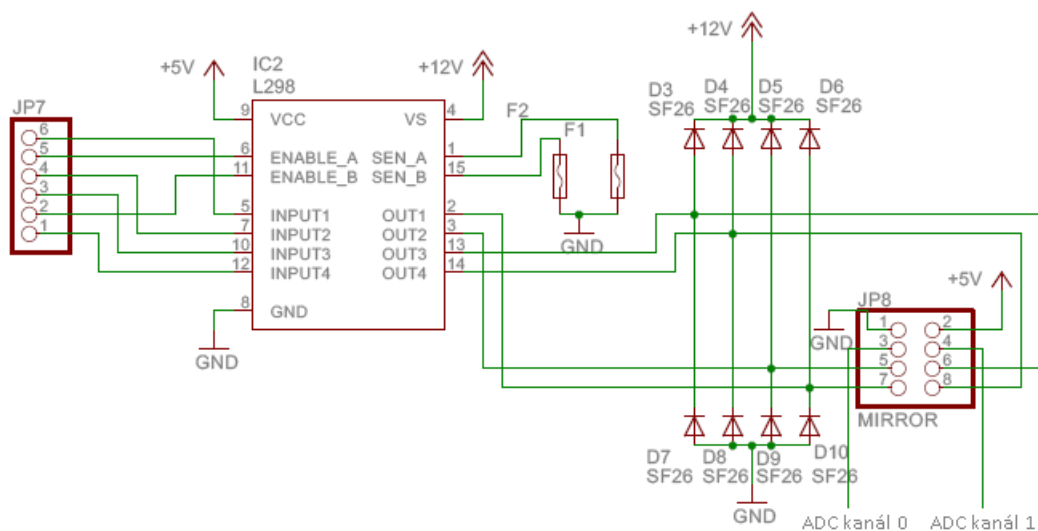
6.4.3 Spínání motorů

Pro spínání stejnosměrných motorů na 12 V byl vybrán obvod L298, který obsahuje dva plné H-můstky. Obvod pro svoji funkčnost potřebuje napájecí napětí pro logické obvody +5 V a napájení pro připojené motory v rozsahu 7,5 V až 46 V. Maximální spínaný proud je 2 A. Ovládání jednoho můstku je pomocí tří signálů, viz tabulka .

Tabulka 3: Ovládání motoru H-můstkem

Enable	IN1	IN2	Motor
H	L	L	Motor zkratován
H	H	L	Otáčení vpravo
H	L	H	Otáčení vlevo
H	H	H	Motor zkratován
L	X	X	Motor odpojen

Mezi výstupem z můstku a motorem jsou ještě připojeny diody sloužící k ochraně obvodu při vypnutí motorů (viz Obrázek 37).



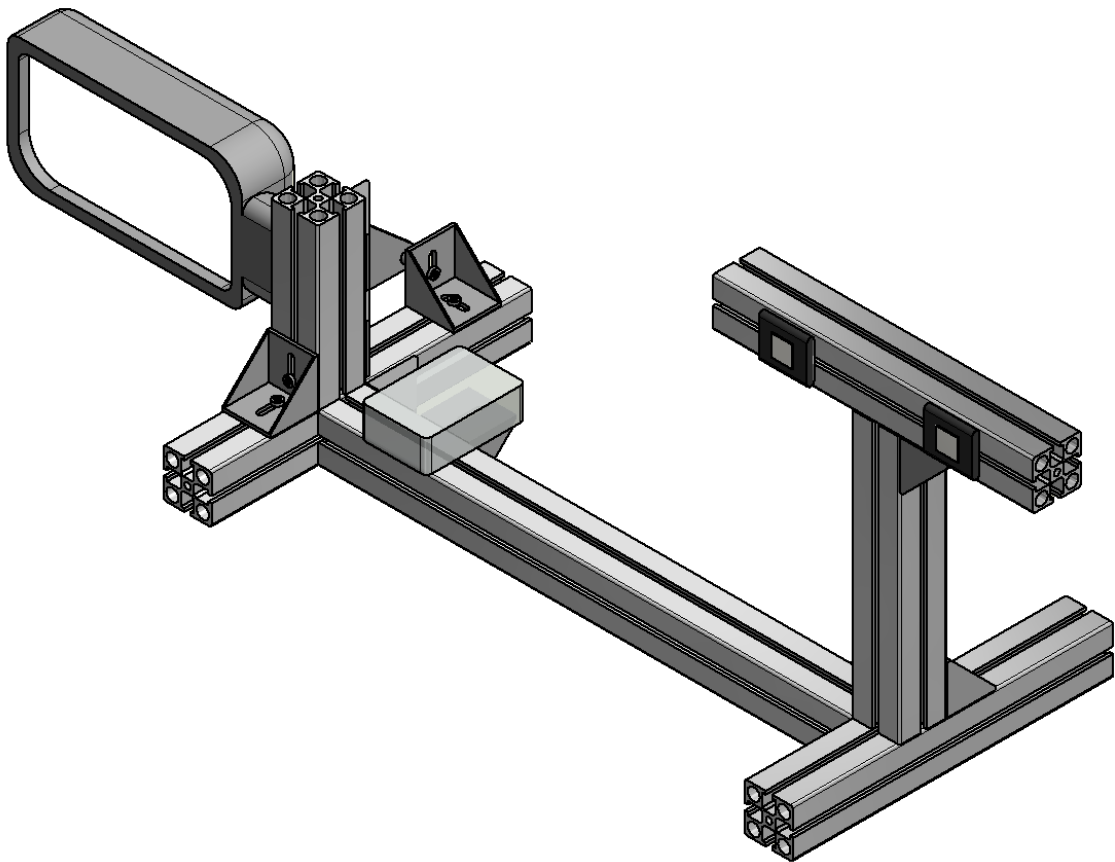
Obrázek 37: Zapojení dvojitého H-můstku pro řízení natočení zrcátka

6.5 Návrh stojanu

Z důvodu potřeby vývoje a testování systému byl navržen stojan, na který lze připevnit všechny součásti zařízení a které simuluje podmínky v automobilu; především rozmístění kamer a zrcátka odpovídá rozměrům uvedeným v popisu scény (str. 29).

Konstrukce je navržena z hliníkových profilů (45x45) mm. Tento montážní systém nabízí variabilitu a není problém v případě nutnosti konstrukci jakkoliv upravit.

Návrh konstrukce a podklady pro potřebný materiál byly provedeny v programu Autodesk Inventor.



Obrázek 38: Konstrukční 3D návrh stojanu pro testování zařízení

7 SOFTWAREVÁ IMPLEMENTACE

Tato kapitola se zabývá popisem dostupných softwarových prostředků a ve větší části pak softwarovou implementací algoritmů.

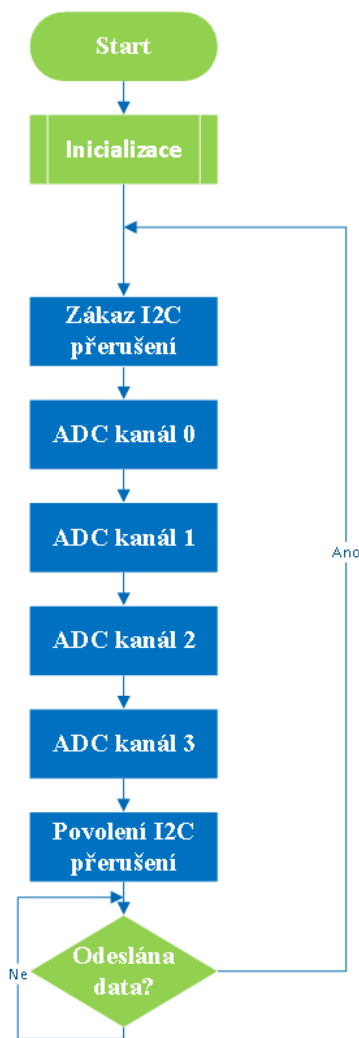
7.1 ATMega8

Program pro ATMegu byl napsán ve vývojovém prostředí Atmel Studio 6.2, které je založené na platformě Visual Studio Shell. Atmel Studio nabízí podporu knihoven pro široký záběr mikrokontrolérů právě od firmy Atmel. Studio ovšem nepodporuje použitý programátor USBasp. Pro nahrání programu do procesoru byl použit program eXtremeBurner - AVR. Programátor využívá sériového programování 8bitových čipů od Atmelu.

Tento modul má za úkol cyklicky provádět AD převod na čtyřech kanálech a na vyžádání data odeslat přes I2C komunikaci. Program využívá integrované nastavitelné přerušování. První přerušování nastává v okamžiku dokončení převodu, kdy se signalizuje, že je možné data vyčíst, přepnout kanál převodníku a opět spustit AD převod. Druhým přerušování způsobuje příjem zprávy po I2C. Respektive detekce adresy daného zařízení. Po přijetí požadavku na vyčtení, procesor odešle přepravená data. Celé chování programu je více patrné z vývojového diagramu. Měření pozice zrcátka probíhá s 10-ti bitovou přesností, zatímco na vyhodnocení ovládání postačuje rozlišení 8 bitů.

Při současném povolení přerušování od obou zdrojů docházelo k odesílání špatných dat. K tomu docházelo díky zpoždění mezi přijatou žádostí o data od mastera a následným odesláním dat. Pokud se mikroprocesor v době požadavku na data nacházel v obsluze AD převodníku, nestihl včas zareagovat. Tato kolize je vyřešena zakázáním přerušování v době průběhu převodu. Po změření všech čtyř kanálů program čeká, dokud nebudou všechny data odeslána.

Výhodou využití mikroprocesoru oproti samostatnému AD převodníku je možnost předzpracování změřených dat. Toho je využito u vyhodnocení ovládání zrcátka. V případě přepínače levá/pravá, jsou vyhodnocovány tři možné stavy. Pozice levé zrcátka, kdy je přepnuto na odpor $R = 0,7 \text{ k}\Omega$.



Obrázek 39: Algoritmus AD převodu a odesílání dat přes I2C

Hodnota reprezentována v rozsahu 0 až 255 je dle rovnice (7-1) rovna číslu 105. Při přepnutí na pozici ovládání pravého zrcátka ($R = 1,5 \text{ k}\Omega$) odpovídá převedená hodnota číslu 153. Při nastavení přepínače do neutrální polohy – automatické řízení, je obvod rozeznut a hodnota napětí na vstupu je rovna napětí napájecímu.

$$C = 255 * \frac{R_x}{1k + R_x} \quad (7-1)$$

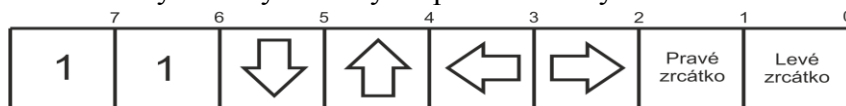
Ze známých hodnot lze v programu provést předzpracování prostým prahováním změřené hodnoty. Prahy jsou určeny v polovině rozsahu mezi sousedními možnostmi.

Tabulka 4: Nastavené prahey pro určení stavu ovladače zrcátka

Prahované stavy		Práh
levé zrcátko	pravé zrcátko	129
pravé zrcátko	automat	204
doleva	doprava	33
doprava	nahoru	90
nahoru	dolu	151
dolu	-	220

Po zpracování je hodnota posílána jako nejnižší dva bity proměnné Joystick, která udává stav ovládání. Nultý bit značí levé tlačítko, první bit pravé. Pokud je zvolen automatický režim jsou nastaveny oba bity do jedničky. Zbylé bity této proměnné jsou použity pro další vyhodnocení zrcátka.

Stejným způsobem jsou předzpracována data ze čtvrtého kanálu převodníku. Tedy hodnota udávající pozici ovládání, udávající směr pohybu zrcátka. Tato hodnota je narahována dle tabulky 4 a bity uloženy do proměnné Joystick.



Obrázek 40: Funkce jednotlivých bitů v proměnné Joystick

Data odesílaná přes I2C do Raspberry obsahují šest bajtů. První bajt obsahuje značku začátku přenosu. V tomto případě hodnotu 0xBB. Dle této definované hodnoty lze synchronizovat komunikaci na obou stranách. Druhý a třetí bajt obsahují hodnotu horizontálního snímače natočení, čtvrtý a pátý hodnotu vertikálního snímače natočení a poslední bajt udává stav ovládání zrcátka.

Tabulka 5: Pořadí odesílání bajtů přes I2C

pořadí bajtu	přenášená informace
1.	Synchronní znak 0xBB
2.	Horizontální poloha - dolních 8 bitů
3.	Horizontální poloha - horní 2 bity
4.	Vertikální poloha - dolních 8 bitů
5.	Vertikální poloha - horní 2 bity
6.	Joystick - stav ovladače zrcátka

7.2 Instalace Raspbian

Z dostupných operačních systémů RPi 2 je pro tento projekt zajímavá distribuce Linuxu s názvem Raspbian. Je to neoficiální distribuce operačního systému založeného na Linuxu Debian Wheezy, ale s nastavením kompilace pro hardware Raspberry. Všechny dostupné operační systémy lze stáhnout z oficiálních stránek. Druhou možností je stažení operačního systému NOOBS, který slouží jako zavaděč pro instalaci uvedených operačních systémů.

Tabulka 6: Dostupné operační systémy pro Raspberry Pi 2

Raspbian – Debian Wheezy
Ubuntu MATE
Snappy Ubuntu Core
Windows 10 IoT Core
OSMC
OpenELEC
PiNet
RISCOS

Raspberry Pi 2 bylo dodáno i s kartou obsahující operační systém Raspbian. Problém nastal v případě použití 8 GB karty, na které zbýval pouze 1 GB volného prostoru. Pro instalaci vývojového prostředí a potřebných knihoven je k zapotřebí alespoň 3 GB místa v paměti. Problém byl vyřešen opětovnou instalací operačního systému, kdy byl zvolen stejný operační systém Raspbian, ovšem bez dalších doplňujících aplikací. V tomto případě nebyl vytvořen na kartě obraz zálohy systému a výsledné celkové místo k dispozici pro uživatele dosáhlo hodnoty 6 GB.

Model Pi 2 dosahuje již dostatečného výkonu k možnosti vytvářet aplikace přímo na něm. K vývoji aplikace v programovacím jazyce C++ bylo zvoleno vývojové prostředí QT Creator.

7.3 Použité knihovny

V systému Linux se nabízí využít knihovnu podporující počítačové vidění OpenCV – Open Source Computer Vision. Knihovna podporuje programovací jazyky C, C++, Java a Python na platformách Windows, Linux, iOS a Android.

Knihovna obsahuje více modulů sjednocujících funkce dle jejich použití. Knihovna poskytuje kompletní podporu pro nastavení a ovládání kamer, uživatelské rozhraní včetně zobrazování snímků, předzpracování obrazu a zpracování videa. Dále obsahuje moduly určené ke specifickým účelům, jako je rozpoznávání objektů v obraze, kalibrace kamery, použití 3D rekonstrukce obrazu, strojové učení, použití neuronových sítí pro zpracování obrazu a další.

V tomto projektu je použita nejnovější verze dostupná při jeho tvorbě; verze 3.1 vydaná v prosinci 2015.

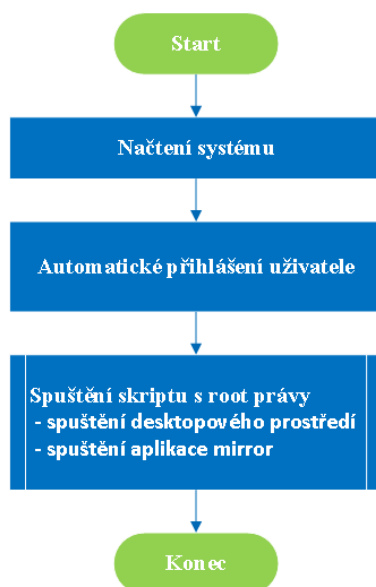
K přístupu k GPIO konektoru při programování v jazyce C/C++ slouží knihovna WiringPI. Tato knihovna je volně stažitelná z internetu a nabízí funkce pro nastavení a používání jednotlivých pinů. Knihovna podporuje snadné využití zabudovaných komunikací, kdy pomocí dvou funkcí lze jednoduše používat I2C sběrnici. Pro přístup k pinům ovšem knihovna potřebuje práva uživatele root. Aplikaci musí spustit uživatel root anebo jiný uživatel s použitím příkazu `sudo`, opravňujícího aplikaci spustit s právy uživatele root.

7.4 Spuštění programu

V reálném použití zařízení v automobilu je vyžadováno automatické spuštění řídicí aplikace po startu automobilu. Pro softwarovou implementaci tento požadavek znamená, že aplikace se musí automaticky spustit po startu systému Raspbian.

Splnění požadavku lze zajistit napsáním skriptu pro spuštění programu a jeho automatické spuštění při startu systému. Jednou z možností je nahrání skriptu do adresáře `/etc/Init.d`, ze kterého systém spouští skripty při startu systému. Zde nastává problém s pořadím spouštění, aby se vlastní skript provedl až po načtení zbytku systému. Nebo lze připsat příkaz na spuštění požadovaného skriptu do souboru `.profile`, který se spouští až po přihlášení uživatele. V takovémto případě je zajištěno, že se program spustí až po načtení všech částí operačního systému.

Aby byl splněn požadavek na spuštění aplikace s právy uživatele root, je spuštění grafické nadstavby systému provedeno s příkazem `sudo`. Aplikace, která se spustí v zápětí, se již zapne s těmito právy.



Obrázek 41: Postup automatického spuštění při zapnutí Raspberry

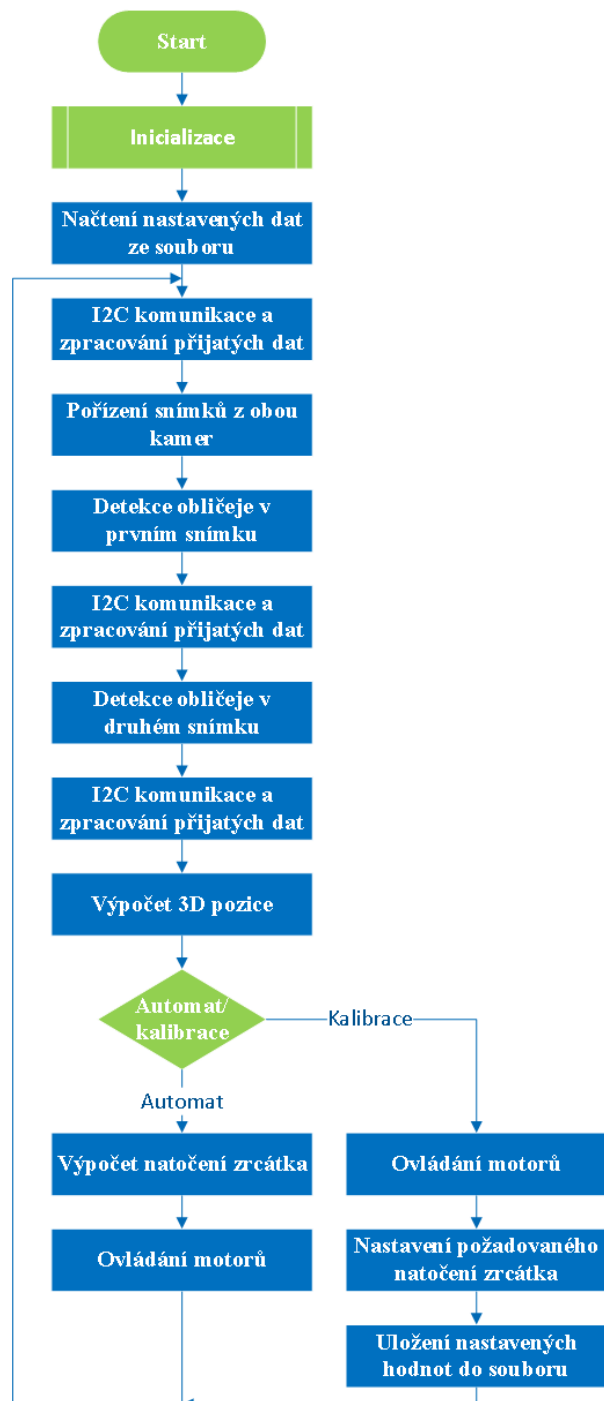
7.5 Architektura aplikace

Vytvořená aplikace má za úkol následující postup: načíst data pozice zrcátka a stav ovládání přes I2C, detekovat tvář v obou snímcích a v režimu automat vypočítat změnu polohy zrcátka a řídit motory. Nebo v manuálním režimu nastavit polohu zrcátek a uložit data o nastaveném vztahu mezi polohou obličeje a natočením zrcátka. Celkové chování programu nejlépe vystihuje vývojový diagram na následujícím obrázku.

Pokud má zařízení fungovat po startu, bez opětovné kalibrace zrcátka, je důležité nastavené parametry uložit a při startu programu načíst. K účelu uložení dat se používá textový soubor `Set_values`. Zápis, respektive čtení, se provádí v binárním režimu, kdy není nutné čísla k uložení převádět na textový formát, ale lze je zapsat, respektive načíst, přímo.

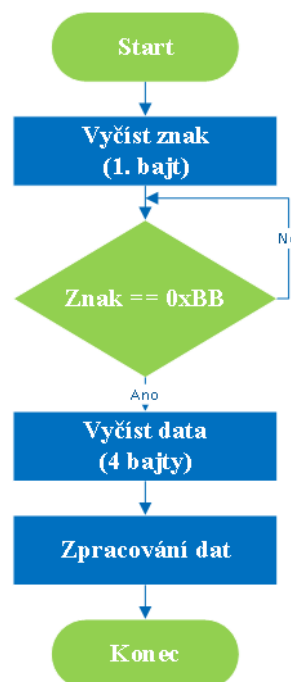
Komunikace I2C pro získání stavu zrcátka a ovladače probíhá v programu třikrát. Tím se zkrátí reakční doba programu na ovládání motorů v automatickém i manuálním režimu z původních 210 ms na 70 ms.

Aplikace je naprogramována s využitím již zmíněných knihoven pro zpracování obrazu a pro použití digitálních vstupů/výstupů. Ovládání zrcátka zastřešuje třída `cMirror`, které se při inicializaci předají čísla pinů, na které jsou připojeny můstky pro řízení motorů. Třída obsahuje metody pro spínání jednotlivých směrů pohybu. Vzhledem k zapojení motorů v zrcátku se společným pinem (Obrázek 28), není možné sepnout motory naráz v opačném směru. Metody pro řízení motorů umožňují spustit pouze jedem motor. Spuštění druhého motoru způsobí zastavení předchozího.



Obrázek 42: Vývojový diagram znázorňující architekturu vytvořené aplikace

Povely pro program jsou přijímány přes I2C komunikaci. Raspberry je v roli mastera, kdy vysílá požadavek na data a ATmega na požadavek odpovídá odesláním připravených dat. Synchronizace zařízení je řešena identifikačním znakem, který se odesílá jako první. Teprve pokud znak souhlasí, jsou přijata platná data, viz Obrázek 43. Data jsou následně zpracována. Přijaté bajty o poloze zrcátka jsou posílány jako čtyři bajty, protože jsou v 10-ti bitové přesnosti. Z dat je dále vyhodnocen přepínač ovládání zrcátka a dle jeho polohy se řídí stav programu automat/kalibrace, případně manuální ovládání zrcátka.



Obrázek 43: Postup I2C komunikace

7.6 Nastavení kamer

Důležitým krokem v aplikaci je inicializace a nastavení kamer. Pokud je některá z kamer nedostupná, aplikace se ukončí chybovým hlášením. V nastavení kamer je nejdůležitější parametr rozlišení výstupu. Pro použité kamery lze nastavit rozlišení 640x480 px, nebo 320x240 px. Vyšší rozlišení znamená vyšší citlivost v lokalizaci obličeje, avšak za cenu delšího času zpracování, způsobeného především výrazným zpomalením detekce obličeje. Srovnání časů obou nastavení je v tabulce 7 Tabulka 7.

Tabulka 7: Srovnání potřebného času k provedení jednoho cyklu algoritmu dle zvoleného rozlišení kamery

Rozlišení [px]	Perioda zpracování [ms]
640x480	370
320x240	210

Pro aplikaci bylo vybráno s ohledem na dobu zpracování rozlišení 320x240 pixelů.

Další nastavitelné parametry i s jejich nastavenými hodnotami jsou uvedeny v následující tabulce.

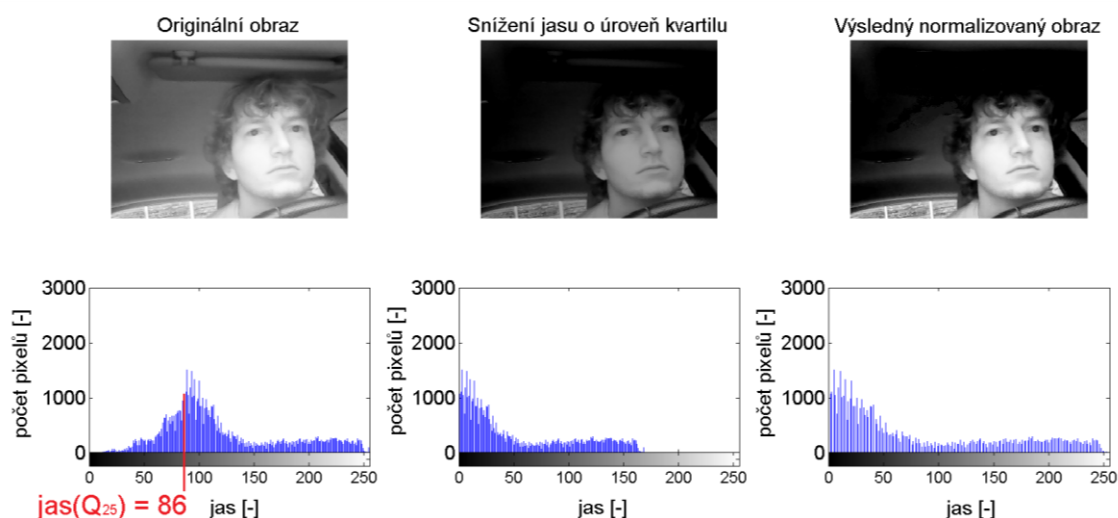
Tabulka 8: Nastavitelné parametry kamery

Parametr	Rozsah	Nastavená hodnota	Popis
FRAME_WIDTH	640, 320	320	
FRAME_HEIGHT	480, 240	240	
GAIN	0 až 1	1	Zesílení jde sice nastavit, avšak kamera ho sama koriguje dle světelných podmínek
SATURATION	0 až 1	0	Nastavení saturace nemá vliv pro dané podmínky s použitím filtru
MODE	0, 1	2	výstupní barevný formát 0=BGR, 1=RGB, 2=grayscale, 3=YUYV
CONTRAST	0 až 1	1	

7.7 Předzpracování obrazu

Vybraný detektor popsany dále pracuje s monochromatickým obrazem. Proto je výstupní obraz z kamery nejprve ze všeho převeden z RGB na šedotónový obraz. Dále v programu se již pracuje pouze s tímto obrazem.

Druhou použitou částí předzpracování obrazu je jasová transformace. Důvodem je zvýšení kontrastu pořízených snímků. Využívá se skutečnosti, že obličej se v definované scéně nebude nacházet v dolní čtvrtině jasové stupnice. Předzpracování nejprve zjistí do jaké úrovně se nachází dolní kvartil (Q_{25}) všech pixelů. O tuto jasovou úroveň sníží jas celého obrazu. Další výhodou je zamaskování objektů nacházejících se v tomto rozsahu jasů. Jednolitější pozadí přispívá k přesnější detekci. Následně dojde k normalizaci – lineární roztažení jasových hodnot do celého rozsahu stupnice. Obrázek níže popisuje použitý postup předzpracování.



Obrázek 44: Použitý postup předzpracování obrazu

Třetím krokem předzpracování vstupního obrazu se nabízí filtrace. Ovšem z podstaty detektoru Viola-Jones, který porovnává oblasti pixelů tento krok nemá žádný vliv na zlepšení přesnosti detektoru. Předzpracování by zabralo více času bez užitku ke kvalitě algoritmu. Proto vstupní obrázek není v předzpracování filtrován.

7.8 Detektor obličeje

K vybrání vhodného detektoru obličeje musíme nejdříve specifikovat požadavky, které budou na detektor v dané aplikaci kladeny. Pro tento případ to jsou:

- Jednoduchá implementace bez nutnosti databáze při klasifikaci
- Správná funkčnost přes den i v noci
- Detekce i mírně natočeného obličeje
- Perioda detekce maximálně 0,5s (navrženo, nutno experimentálně ověřit)

Použití IR přísvitu a optického filtru způsobilo ztrátu barevné informace. K detekci tedy nemůžeme využít detekci dle barvy kůže, která by mohla detekci zrychlit. K detekci byl vybrán často používaný detektor obličejů Viola-Jones. Algoritmus detekce a jeho jsou uvedeny v kapitole 3.6.

K vytvoření detektoru musíme nejprve definovat podmínky detekce. V první řadě především minimální velikost obličeje, který má být nalezen. Dle toho se nastaví počáteční velikost podokna. Z experimentu bylo zjištěno, že při rozlišení kamery 320x240 má ve vzdálenosti 1,5m obličej řidiče velikost přibližně 65x70 pixelů. Záleží nejen na konkrétním řidiči, ale také na umístění kamery v automobilu. Minimální velikost podokna byla zvolena 60x60 px. V praxi se používají podokna až o velikosti 24x24 px, ale z důvodu potřeby většího rozlišení pro přesnější určení pozice obličeje, nemůžeme obraz takto podvzorkovat.

Při psaní programu bylo využito knihovny OpenCV a použití detektoru právě z této knihovny. Výhodou je ušetření času s programováním vlastního detektoru a použití ozkoušeného a vyladěného algoritmu. V OpenCV slouží k použití detektoru Viola-Jones metoda `detectMultiScale` třídy `CascadeClassifier` nacházející se v modulu `objdetect`. Vstupní proměnnou je obraz, ve kterém chceme objekty detekovat. Výstupem z funkce je seznam detekovaných objektů, udávající jejich pozici a velikost.

Tato funkce umožňuje nastavení následujících parametrů detekce:

- minimální velikost objektu
- maximální velikost objektu
- změna velikosti obrazu v každém cyklu
- počet nutných sousedících detekovaných objektů

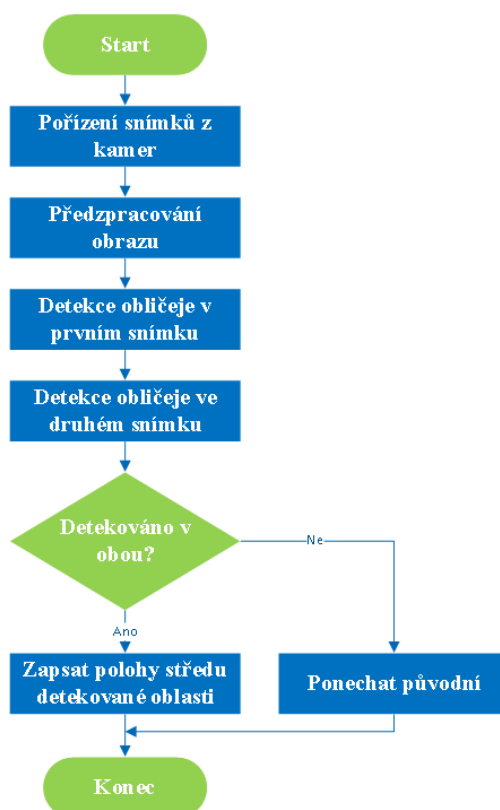
Maximální velikost objektu je nastavena na polovinu velikosti obrázku. Větší objekty je zbytečné detekovat, protože v tom případě nebudou současně v záběru obou kamer.

Této velikosti lze dosáhnout ve vzdálenosti 50 cm od kamer. Nejmenší velikost detekovaného objektu je nastavena na 60x60 pixelů, jak je popsáno výše.

Změna velikosti obrazu udává kolikrát se v každém kroku zmenší obraz ve kterém se detekuje. To odpovídá změně kroku velikosti detekovaného objektu. Tento parametr byl zvolen 1,2 s ohledem na rychlost detektoru.

Poslední parametr udává kolik sousedních objektů zahrnuje jeden výsledný detekovaný objekt. Toto vychází ze skutečnosti, že detektorem Viola-Jones jsou objekty detekovány duplicitně vícekrát. Nastavením vhodné hodnoty se zamezí duplicitní detekci jednoho objektu a navíc se tím detektor stává robustnější. Každý objekt musí být detekován vícekrát, aby byl uznán za hledaný objekt. Experimentálně byl tento parametr nastaven na hodnotu 5.

Knihovna OpenCV obsahuje sadu již natrénovaných detektorů uložených v .xml souborech. Obsahují detektory obličeje, očí, celého těla a dalších. Pro detekci obličeje zde figurují 3 detektory: `frontal_face_alt.xml`, `frontal_face_alt2.xml` a `frontal_face_default.xml`. V projektu byly vyzkoušeny všechny tři možnosti. Výsledky použití budou uvedeny v kapitole 8.1.2.



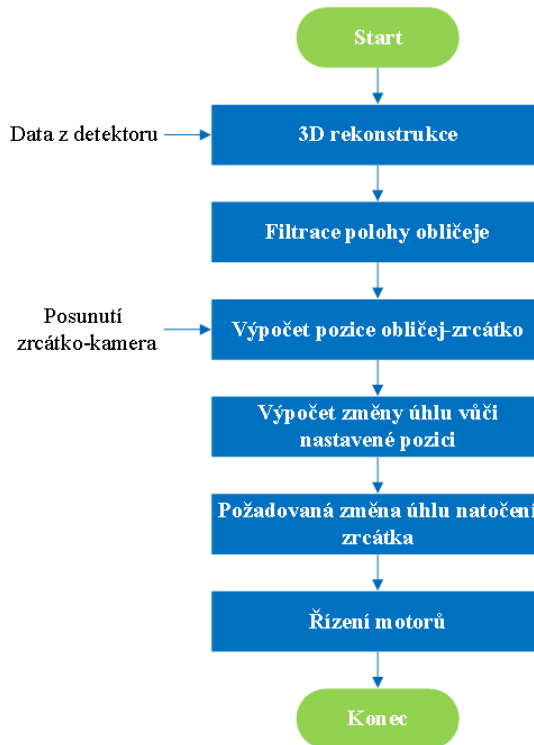
Obrázek 45: Algoritmus implementované detekce obličeje

Na předchozím vývojovém diagramu lze vidět přesný algoritmus použitý při detekci obličeje ve snímcích z obou kamer. Poloha pozice obličeje ve snímcích je aktualizována pouze v případě, kdy je obličej detekován na obou snímcích.

Z výstupního seznamu nalezených tváří je vybrána první a z detekovaných rozměrů vypočítán střed tváře.

7.9 Výpočet natočení zrcátka

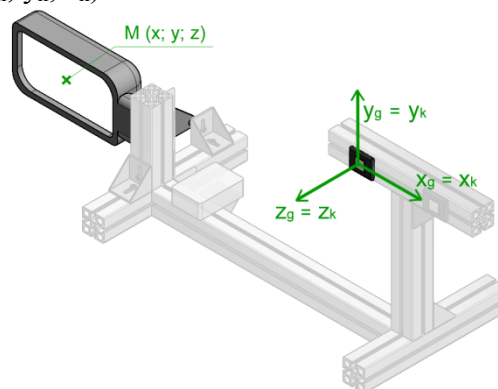
Algoritmus automatického natočení zrcátka využívá znalost pozice obličeje vůči zrcátku a současně pozice zrcátka. Ze změny polohy tváře lze vypočítat potřebné pootočení zrcátka, aby výhled stále odpovídal nadefinované scéně.



Obrázek 46: Postup výpočtu automatického natočení zrcátka

7.9.1 Souřadnicové systémy

Pro výpočet požadovaného natočení zrcátka a s tím související pozice obličeje vůči zrcátku, musí být nejdříve definovaný souřadný systém, vůči kterému budou počítány veškeré posuvy. Situace lze shrnout do dvou souřadnicových systémů nepočítáme-li obrazové souřadnice; globální souřadnicový systém $S_g(x_g, y_g, z_g)$ a souřadnicový systém stereovize – kamer $S_k(x_k, y_k, z_k)$.



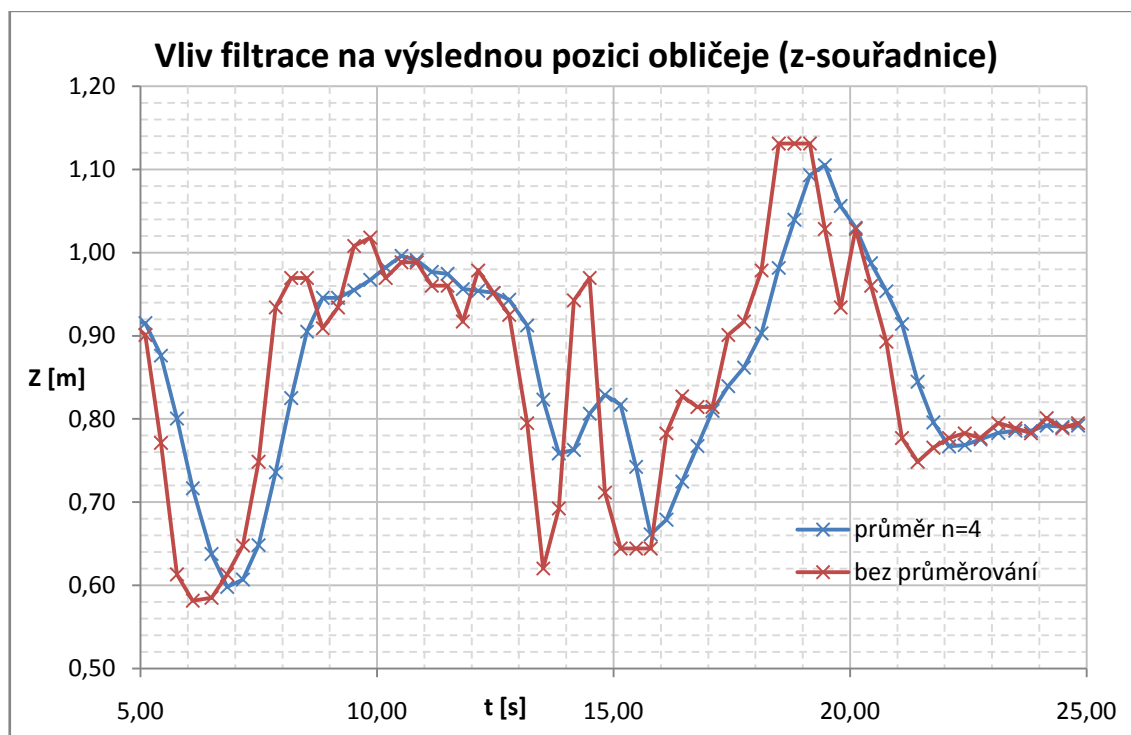
Obrázek 47: Vyznačení zvolených souřadnicových systémů použitých pro výpočty

Výpočet 3D souřadnic ze souřadnic obrazových probíhá dle vztahů uvedených v kapitole 4.2. Tyto souřadnice jsou počítány vůči souřadnicovému systému levé kamery, který je shodný se souřadnicovým systémem stereovize S_k . Pro zjednodušení následného přepočítávání jsou osy globálních souřadnic S_g shodné s osami tohoto systému S_k .

Pro výpočet je důležité definovat souřadnice zrcátka $M(x, y, z)$. Z těchto souřadnic a ze známých souřadnic pozice obličeje lze spočítat vzájemnou pozici těchto bodů.

7.9.2 Filtrace polohy obličeje

Poloha obličeje ve 3D souřadnicích je filtrována pomocí klouzavého průměru. Filtrací jsou potlačeny rychlé změny polohy a nepřesnost detektoru, kdy na po dvou po sobě jdoucích snímcích se stejnou pozicí obličeje detektor neurčí pozici objektu vždy stejně. Výpočet průměrů byl vyzkoušen pro $n=3$, 4 a 5. Odezvy z jednotlivých měření jsou uvedeny v příloze D. Experimentálně bylo jako nejvhodnější určeno $n=4$. Pro tuto hodnotu se při skokové změně polohy filtrovaná pozice vyrovná detekované po 0,9s. Rychlé změny a záškuby jsou potlačeny a zrcátko zbytečně nekmitá. Srovnání průběhů filtrované a nefiltrované složky lze vidět v grafu na obrázku níže.



Obrázek 48: Vliv filtrace na výslednou pozici obličeje

7.9.3 Úhel natočení zrcátka

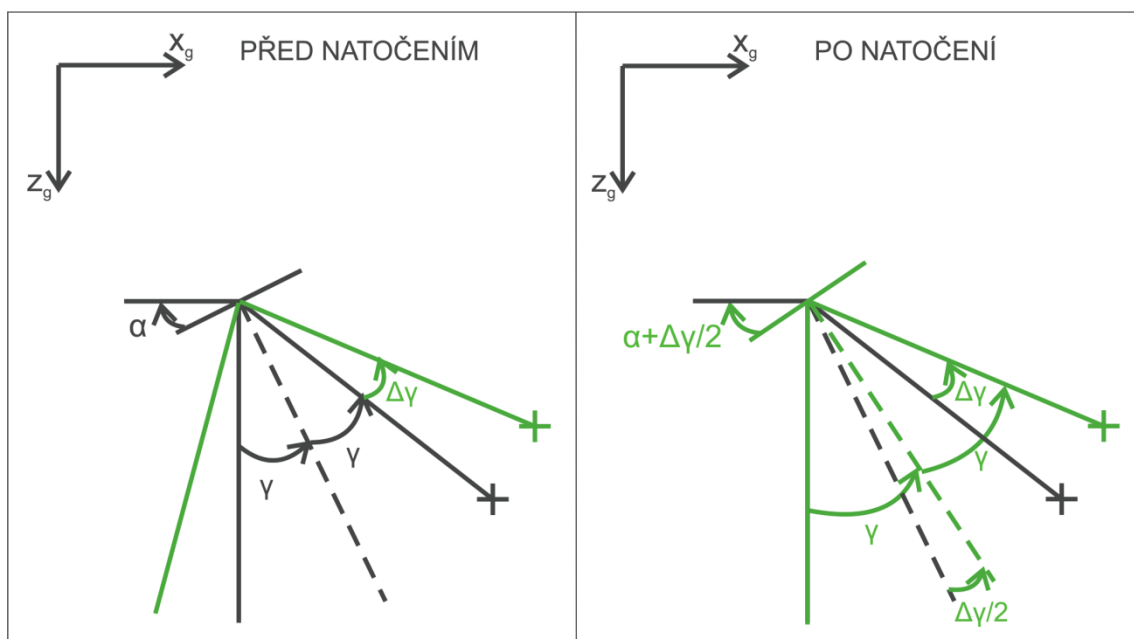
Změřený rozsah natočení použitého zrcátka je roven $\alpha=21^\circ$ v horizontálním směru a $\beta=20^\circ$ ve směru vertikálním. Řízení zrcátek probíhá v rozsahu změřených 10-ti bitových hodnot, zatímco výpočet požadovaného natočení probíhá ve stupních. Přepočet mezi těmito hodnotami je dán rovnicí (7-2). Kde x je vstupní veličina, y výstupní a zbylé proměnné udávají rozsah jednotlivých veličin.

$$y = y_{\min} + (x - x_{\min}) * \frac{y_{\max} - y_{\min}}{x_{\max} - x_{\min}} \quad (7-2)$$

Při konstatování, že úhel odrazu se rovná úhlu dopadu, můžeme požadovaný úhel natočení α_{k+1} spočítat ze vztahu

$$\alpha_{k+1} = \alpha_k + \frac{\Delta\gamma}{2} \quad (7-3)$$

kde γ udává úhel mezi obličejem a zrcátkem. Pro přehlednost je situace znázorněna níže. Princip je vysvětlen na 2D nákresu, ale stejná zákonitost platí i pro vertikální směr. Tím je zařízení schopno zrcátkem sledovat polohu obličeje ve 3D prostoru.



Obrázek 49: Změna natočení zrcátka při změně polohy obličeje

Pokud natočení zrcátka ve vertikálním směru označíme β a úhel obličeje vůči zrcátku v tomto směru δ , jsou výsledné přepočtení vztahy následující:

$$\Delta\gamma = \tan^{-1}\left(\frac{x}{z}\right) - \gamma_s, \quad (7-4)$$

$$\Delta\delta = \tan^{-1}\left(\frac{y}{z}\right) - \delta_s, \quad (7-5)$$

$$\alpha = \alpha_s + \frac{\Delta\gamma}{2} \text{ a} \quad (7-6)$$

$$\beta = \beta_s + \frac{\Delta\delta}{2}. \quad (7-7)$$

Kde index s označuje nastavené hodnoty.

8 VYHODNOCENÍ CHOVÁNÍ ZAŘÍZENÍ

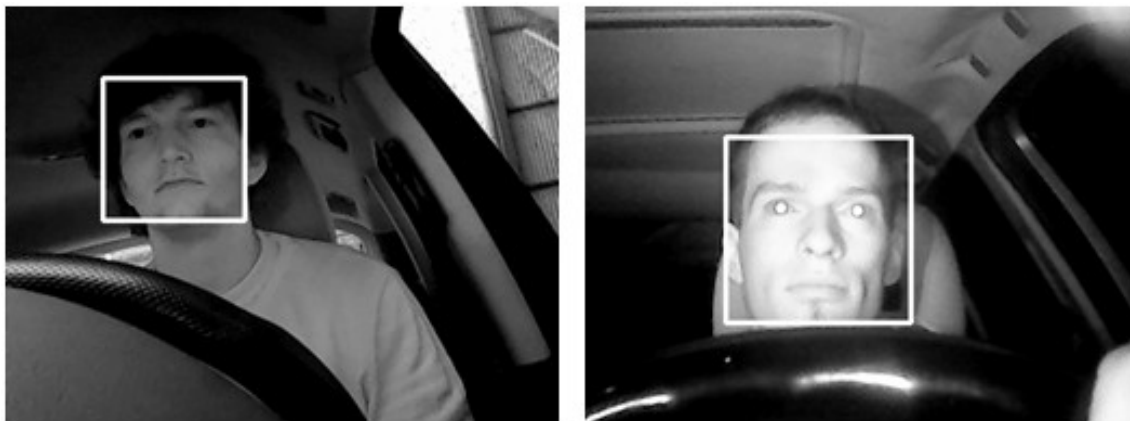
Výsledky vytvořeného zařízení lze hodnotit ze tří hledisek. Prvním faktorem je úspěšnost detekce obličeje v reálných podmínkách, to jest v automobilu. Druhé kritérium vyhodnocení je přesnost měření vzdálenosti pomocí stereovize. Poslední část hodnocení věnující se sledování obličeje zrcátkem přímo závisí na přesnosti měření pozice.

8.1 Úspěšnost detekce obličeje

V následujícím popisu úspěšnosti detekce obličeje, není myšlena úspěšnost detektoru při jednotlivém prohledávání jednoho snímku, ale výsledná úspěšnost detekce na souboru testovacích snímků. Pokud se na testovacím snímku nachází obličej a algoritmus ho detekuje na správném místě, je klasifikace hodnocena jako pozitivní. Správně je také označený negativní výsledek detektoru, pokud se v obraze obličej, či jeho větší část, nenachází. Pokud se ve vstupním obraze obličej nachází, ale detektor ho nerozpozná, či algoritmus označí detekovanou tvář v místě, kde se nenachází jde o nesprávný výsledek.

8.1.1 Testovací soubor

Pro vyhodnocení úspěšnosti detekce byla vytvořena testovací sada snímků, obsahující záznamy z automobilu. V sadě se nachází pozitivní i negativní vzorky. Sada obsahuje snímky ze tří různých automobilů, tří osob a snímky pořízené ve dne a v noci. Příklad těchto snímků je na obrázku dole.



Obrázek 50: Ukázka snímků z vytvořené testovací sady. Snímky jsou po průchodu algoritmem s vyznačeným detekovaným obličejem. Snímek vlevo je pořízen za denního světla, snímek vpravo v noci

8.1.2 Srovnání úspěšnosti použitých detektorů

V projektu byly vyzkoušeny tři různě naučené detektory Viola-Jones. Všechny tři detektory jsou již součástí knihovny OpenCV a jsou určeny pro detekci obličeje. Níže je uvedeno srovnání jejich přesnosti na testovacím souboru.

Tabulka 9: Vyhodnocení přesnosti použitého detektoru "frontalface_alt"

frontalface_alt		Skutečná hodnota	
		Pozitivní	Negativní
Klasifikace	Pozitivní	62	0
	Negativní	53	28

Senzitivita	54%
Specifická	100%
Celková správnost	63%

Tabulka 10: Vyhodnocení přesnosti použitého detektoru "frontalface_alt2"

frontalface_alt2		Skutečná hodnota	
		Pozitivní	Negativní
Klasifikace	Pozitivní	63	0
	Negativní	52	28

Senzitivita	55%
Specifická	100%
Celková správnost	64%

Tabulka 11: Vyhodnocení přesnosti použitého detektoru "frontalface_default"

frontalface_default		Skutečná hodnota	
		Pozitivní	Negativní
Klasifikace	Pozitivní	67	3
	Negativní	48	25

Senzitivita	58%
Specifická	89%
Celková správnost	64%

Nízká čísla senzitivity detektorů jsou dána složením testovací sady. Snímky byly pořízeny tak, aby zachycovaly co nejvíce reálných pozic hlavy řidiče v automobilu. V mnoha případech na snímku není celý obličej, s čímž mají použité detektory problém. Druhým faktorem snižujícím správnost detekce je natočení hlavy směrem od kamery. Detektory zvládají pouze natočení do úhlu přibližně 25°. To je dáno použitou trénovací množinou při učení detektoru.

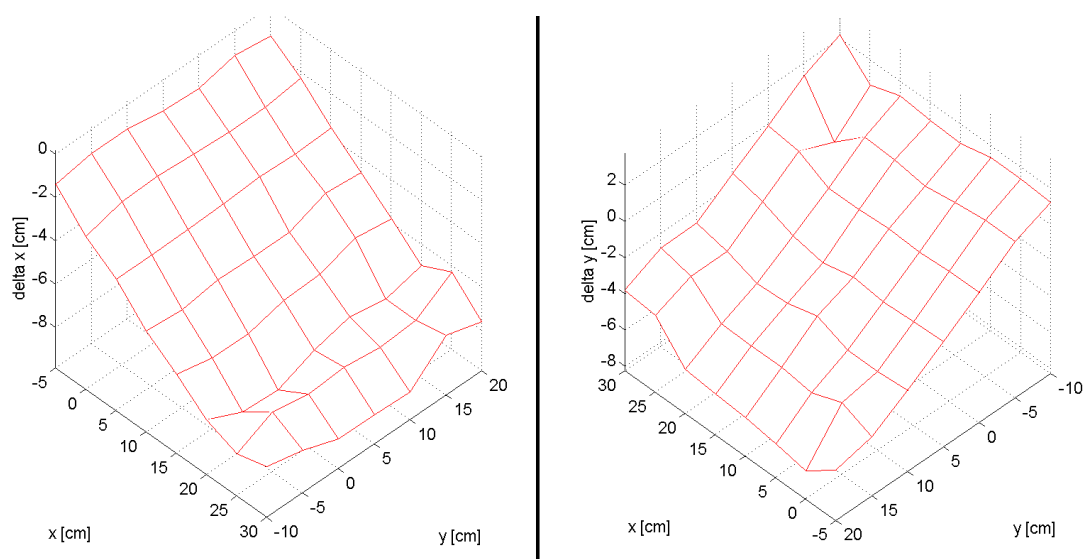
Nejllepší senzitivity dosahoval detektor `frontalface_default`, který zvládl detekovat o 3% více obličejů, než ostatní detektory. Avšak jeho celkovou úspěšnost snižují tři falešné detekce na testovacích datech. Detektory `frontalface_alt` a `frontalface_alt2` mají stoprocentní specifitu.

Z testů nejlépe vyšel detektor `frontalface_alt2`, který dosáhl senzitivity 64%.

8.2 Přesnost měření pozice pomocí stereovize

Testování odchylky chyby měření probíhalo na definované mřížce bodů v rovině XY. Mřížka byla vytvořena s krokem 5 cm v každé ose. Měření probíhalo pro tři zvolené vzdálenosti v ose Z; 60 cm, 90 cm a 120 cm. Pro nastavenou vzdálenost byly změřeny všechny body mřížky, viditelné oběma kamerami. Objekt – obličej se umístil středem do uzlu mřížky, jejíž souřadnice byly brány jako referenční hodnoty pro vyhodnocení.

Pro zvolenou vzdálenost $Z=120$ cm jsou výsledné absolutní chyby měření v osách X a Y vykresleny na následujících snímcích.



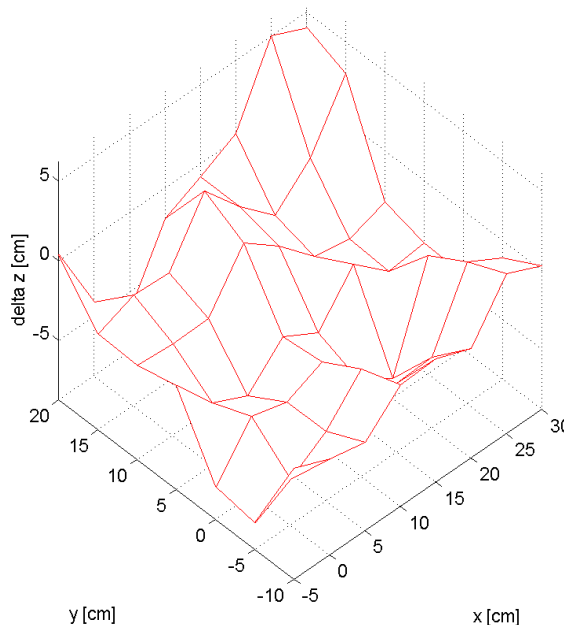
Obrázek 51: Vypočtené absolutní chyby měření polohy v závislosti na měřených souřadnicích. V levé části je zobrazena v měření x-ové souřadnice. Vpravo je vykreslena závislost chyby měření souřadnice y.

Na grafech lze vidět závislost chyby dané souřadnice pouze na velikosti této souřadnice. Na druhé souřadnici je chyba téměř neměnná. Vzhledem k lineární povaze závislosti chyby na velikosti souřadnice lze konstatovat, že v měření ovlivněno systematickou chybou. Výpočet dané souřadnice závisí na obrazové souřadnici a na velikosti paralaxy (viz kapitola 4.2.). Chyba měření je způsobena chybou výpočtu paralaxy. Tato chyba při měření vznikla nepřesným nastavením rovnoběžných optických os kamer.

Z druhého hlediska lze konstatovat, že při dodržení zadaných podmínek stereovize, jsou implementované výpočty měření pozice ve 3D prostoru správné.

Do měření zanáší nepřesnost použitý detektor, který může zareagovat i na malou změnu pozice obličeje větší změnou pozice detekovaného objektu.

Chyba ve výpočtu paralaxy způsobená nepřesným nastavením geometrie stereovize a nejednoznačností detektoru pro stereoskopické snímky zanáší také chybu do vyhodnocení vzdálenosti obličeje od kamery. Tato chyba se pohybovala v rozmezí $\pm 4,2\%$ z měřené vzdálenosti (pro $Z=120$ cm).

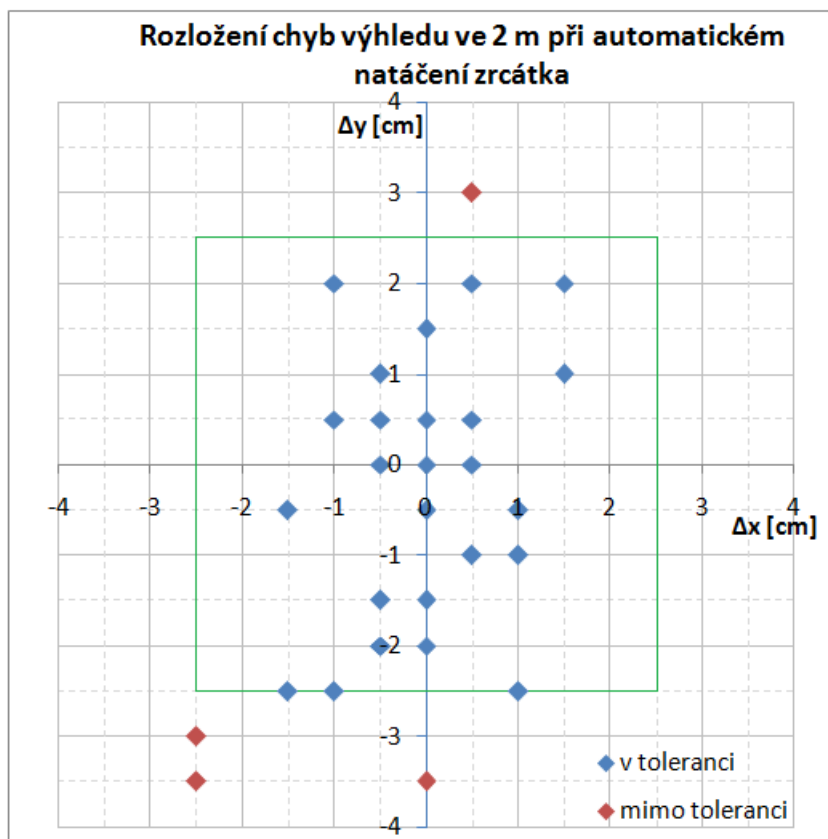


Obrázek 52: Chyba měření vzdálenosti obličeje od kamery pomocí stereovize v závislosti na nastavených souřadnicích x a y

8.3 Přesnost automatického nastavení výhledu v zrcátku

K určení přesnosti automatického natočení zrcátka byl proveden pokus, kdy bylo zrcátko namířeno na čtvercovou síť ve vzdálenosti 2 metrů. Uprostřed zrcátka byl označen bod, který se natočením zrcátka nastavil na střed sítě. Následným přepnutím do automatického režimu zrcátko dle polohy obličeje začalo automaticky nastavovat výhled.

Chyba ve výhledu byla proměřena pro uvedenou výseč stereovize ve vzdálenosti 0,5 až 1,5 metrů. Referenční výhled byl nastaveno pro souřadnice $O_r(0; 0; 1 \text{ m})$. Měření probíhalo s ohledem na odečítání hodnot v referenční síti s rozlišením 0,5 cm. Výsledky měření jsou uvedeny v obrázku 52.



Obrázek 53: Rozložení chyb výhledu ve vzdálenosti 2 m při automatickém natáčení zrcátka. Provedeno 30 měření pro různé polohy obličeje v intervalech uvažovaných v automobilu. Červeně jsou vyznačeny body, které nevyhovují stanovené chybě 5 cm ve vzdálenosti 4 metrů

Z provedených měření 26 vyhovuje stanovené maximální chybě ve výhledu v zrcátku 5 cm ve vzdálenosti 4 metrů od zrcátka. Zbývající 4 měření nevyhovují v ose y. Dle měření je maximální chyba ve vzdálenosti 4 metrů rovna 5 cm v ose x a 7 cm v ose y. Tomu odpovídá absolutní chyba 8,6 cm.

9 ZÁVĚR

V této práci byl popsán návrh zařízení pro automatické natáčení zrcátka. Návrh byl zaměřen především na funkčnost výsledného zařízení. V závislosti na popisu snímané scény byla pro měření polohy v trojrozměrném prostoru vybrána metoda stereovize. Provedením citlivostní analýzy (kapitola 5.3.1) byla vypočtena velikost minimální báze stereovize $b=25$ cm. Pro požadavek funkčnosti ve dne i v noci byla vybrána webkamera A4Tech PK-333 s integrovaným přísvitem, která splňuje i ostatní zadané požadavky. Ke kameře byl přidán optický filtr pásmové propusti odpovídající vlnové délce přísvitu, který pomáhá sjednotit snímky ve dne a v noci.

Pro ovládání bylo vybráno elektricky řízené boční zpětné zrcátko s pamětí, které má již integrované senzory natočení zrcátka. Ke kalibraci zrcátka (nastavení výhledu) a přepínání mezi automatickým režimem a kalibrací bylo použito standardní ovládání zrcátek. K testovacím účelům byl zkonstruován stojan pro všechny součásti, který simuluje rozmístění zrcátka a kamer v automobilu.

Zařízení je založeno na výpočetní jednotce Raspberry Pi 2, na kterém je nainstalován operační systém Raspbian. Použití distribuce Linuxu spojuje výhodu grafického prostředí pro grafické testování řídicí aplikace a výhodu použití knihovny pro zpracování videa OpenCV. Aplikace byla naprogramována v jazyce C++ a je spouštěna automaticky pomocí skriptu při startu Raspberry. Pro projekt byla vytvořena rozšiřující deska, která obsahuje AD převodník pro měření polohy zrcátek v podobě mikroprocesoru ATmega8. Dále deska obsahuje H-můstky pro řízení motorů v zrcátku a část pro rozvod napájení pro motory a řídicí elektroniku.

Pro algoritmus detekce obličeje byl vybrán detektor Viola-Jones, který splňuje všechny definované podmínky. Pro jeho implementaci do programu bylo využito knihovny OpenCV. Byly vyzkoušeny 3 naučené varianty detektoru, které jsou součástí již zmíněné knihovny. Pro ověření a srovnání úspěšnosti použitých detektorů byla vytvořena testovací sada snímků z automobilu. V sadě jsou obsaženy snímky ze tří automobilů, tří různých osob za denního světla a v noci. Výsledky a výběr nejpresnějšího detektoru jsou popsány v kapitole 8.1.2. Z detekované polohy obličeje řídicí program vypočítá souřadnice obličeje v prostoru a dle nich určí požadované natočení zrcátka.

Druhým bodem testování bylo měření přesnosti určení pozice v prostoru. Byla ověřena platnost implementovaných vztahů. Z analýzy průběhu chyby je vyvozeno, že optické osy kamer při pokusu nebyly stoprocentně rovnoběžné, ani při použití vytvořeného stojanu pro zařízení. Vhodným vylepšením zařízení by byla implementace autokalibrace stereovize, ke které již v práci nezbyl prostor.

Funkčnost celého zařízení byla vyzkoušena při měření chyby v automatickém nastavení výhledu. Předem definovaná chyba 5 cm v úrovni zadního rohu automobilu nebyla splněna pro celý rozsah poloh obličeje, neboť ze 30-ti pokusů 4 hodnoty byly mimo vymezený interval. Překročení stanovených mezí bylo ve všech případech změřeno ve vertikální ose natáčení, kde maximální chyba dosáhla hodnoty 7 cm.

Po provedených testech lze zařízení označit za funkční. Nesplnění přesnosti zařízení vychází z již zmíněného nepřesného mechanického uchycení kamer. Vytvořené zařízení slouží jako prototyp k testování algoritmů a přístupů k řešení. Po přizpůsobení konkrétním podmínkám v automobilu lze na této práci založit implementaci automatického polohování zrcátek.

Literatura

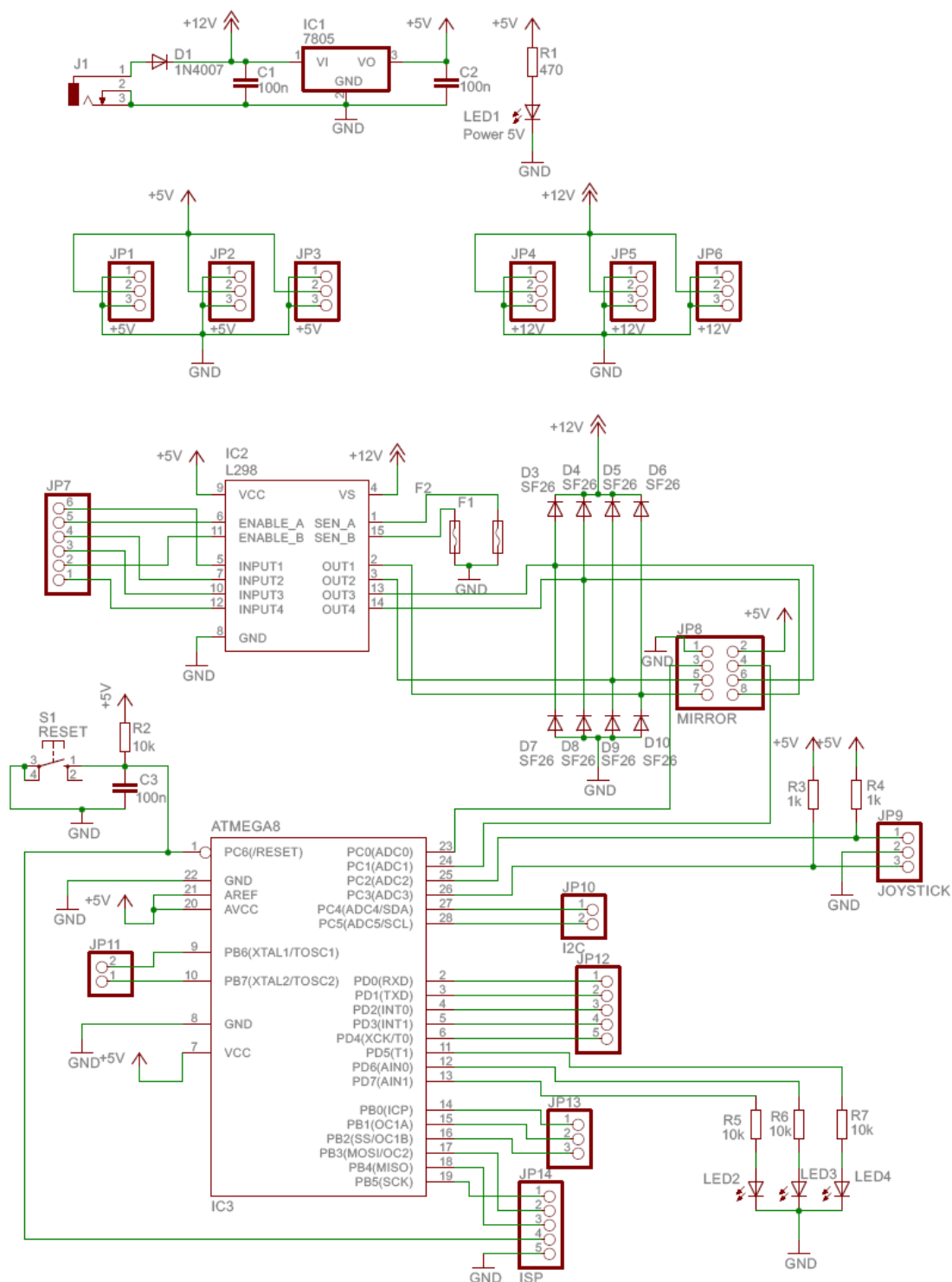
- [1.] MILAN SONKA; VACLAV HLAVAC; ROGER BOYLE. *Image processing, analysis, and machine vision*. 3rd ed., International student edition. Stamford: CengageLearning, 2008. ISBN 04-952-4438-4.
- [2.] DEMANT, Christian, Bernd STREICHER-ABEL a Peter WASZKEWITZ. *Industrial image processing: visual quality control in manufacturing*. Berlin: Springer-Verlag, 1999, 353 s. : il. 1 CD-ROM. ISBN 3540664106.
- [3.] KALOVÁ, Ilona. *Předzpracování obrazu* [prezentace]
- [4.] GONZALEZ, Rafael C. STEVEN L. EDDINS. *Digital image processing: using MATLAB*. Upper Saddle River: PearsonPrenticeHall, 2004, 609 s. ISBN 0130085197.
- [5.] PŘINOSIL, J. *Analýza emocionálních stavů na základě obrazových předloh*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. Vedoucí dizertační práce Mgr. Pavel Rajmic, Ph.D.
- [6.] MING-HSUAN YANG, D.J. KRIEGMAN a N. AHUJA. *Detecting faces in images: a survey*. In: *Pattern Analysis and Machine Intelligence*, IEEE Transactions on. USA: IEEE, 0020n. 1., **24**(1), s. 34-58. DOI: 10.1109/34.982883. ISSN 01628828.
- [7.] VIOLA, Paul a Michael JONES. *Robust Real-Time Face Detection*. In: *International Journal of Computer Vision*. Boston: KluwerAcademicPublishers, 0040n. 1., **57**(2), s. 137-154. DOI: 10.1023/B:VISI.0000013087.49260.fb. ISSN 09205691.
- [8.] JÄHNE, Bernd a Horst HAUßECKER. *Computer vision and applications: a guide for students and practitioners*. San Diego: AcademicPress, 2000, xxi, 679 s. : il. 1 CD-ROM. ISBN 0123797772.
- [9.] SOJKA, Eduard, Jan GAURA a Michal KRUMNIKL. *Matematické základy digitálního zpracování obrazu*. VŠB-TU Ostrava, 2011.

- [10.] (PK-333E) *Clip-on Lighting LED Web Camera*. *A4Tech [online]*. 2015 [cit. 2015-10-13]. Dostupné z: <http://www.a4tech.com/product.asp?cid=1&scid=1&id=593>
- [11.] BROADCOM CORPORATION. *BCM2835 ARM Peripherals: Datasheet*. Cambridge, 2012. Dostupné z: <https://cdn-shop.adafruit.com/product-files/2885/BCM2835Datasheet.pdf>
- [12.] ATMEL CORPORATION. *Datasheet ATmega8-16PU*. San Jose, 2003. Dostupné z: <http://www.gme.cz/img/cache/doc/432/201/atmega8-16pu-datasheet-1.pdf>
- [13.] STMICROELECTRONICS. *2A Positive Voltages regulator: Datasheet*. 2008. Dostupné z: <http://www.gme.cz/img/cache/doc/330/018/78s05-datasheet-1.pdf>
- [14.] Raspberry Pi Documentation. *Raspberry Pi [online]*. 2016 [cit. 2016-05-12]. Dostupné z: <https://www.raspberrypi.org/documentation/>
- [15.] *WiringPi [online]*. 2016 [cit. 2016-05-12]. Dostupné z: <http://wiringpi.com/>
- [16.] OpenCV modules. *OpenCV [online]*. 2016 [cit. 2016-05-05]. Dostupné z: <http://docs.opencv.org/3.1.0/#gsc.tab=0>

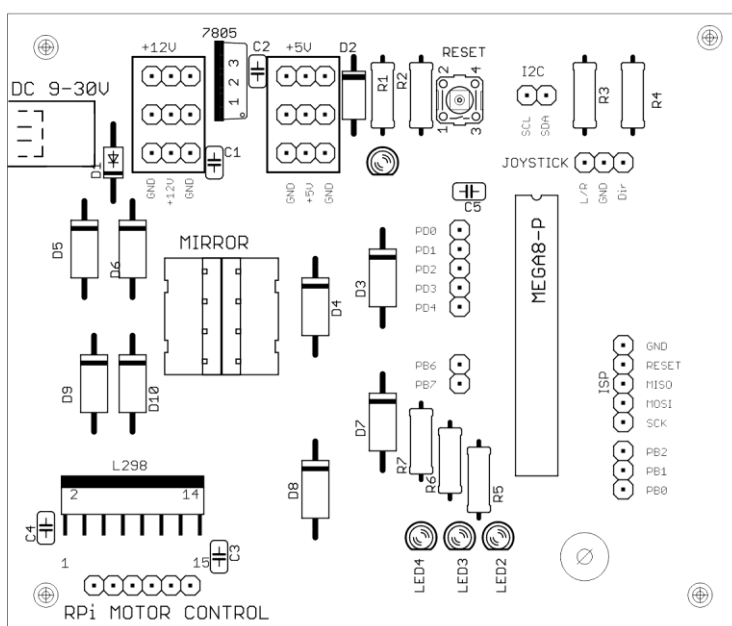
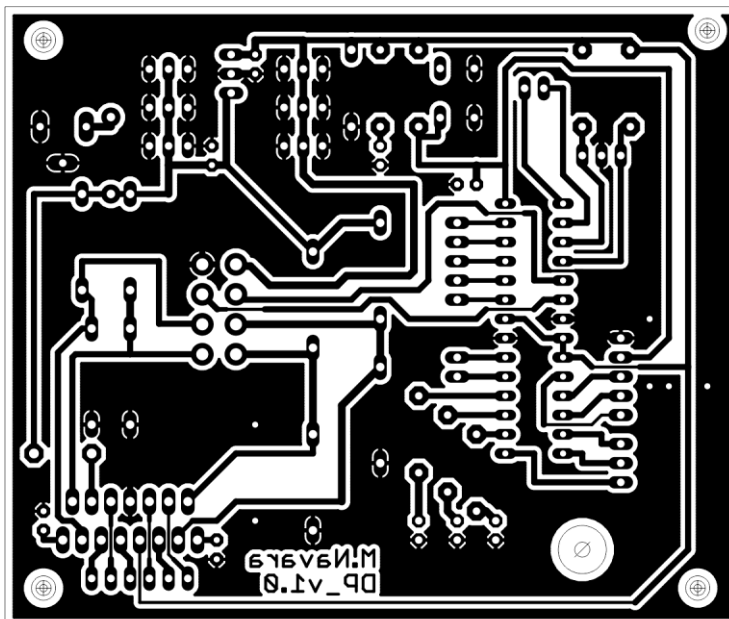
Seznam příloh

- A. Schéma zapojení rozšiřující desky
- B. Podklady pro výrobu rozšiřující desky
- C. Výkresová dokumentace stojanu
- D. Srovnání filtrace polohy obličeje pro různá n
- E. CD
 - E1. Elektronická verze práce
 - E2. Skript pro automatické spuštění programu po startu systému
 - E3. Zdrojové soubory řídicího programu pro Raspberry
 - E4. Zdrojové soubory pro obslužný program ATmega8
 - E5. Testovací sada snímků

Příloha A – Schéma zapojení rozšiřující desky



Příloha B – Podklady pro výrobu rozšiřující desky



Technical drawing of a construction stand (Konstrukce stojanu) showing three views: front, side, and top. Dimensions are provided in millimeters.

- Front View:** Shows the base plate with a width of 300,00 mm and a height of 500,00 mm. The stand is mounted on a base plate with a width of 300,00 mm.
- Side View:** Shows the height of the stand, which is 150,00 mm.
- Top View:** Shows the width of the stand, which is 300,00 mm, and the depth of the base plate, which is 250,00 mm.

The drawing is labeled "Konstrukce stojanu" and "Výkres 1".

Příloha D – Srovnání filtrace polohy obličeje pro různý počet prvků průměru

